IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 26, 2021, accepted October 31, 2021, date of publication November 2, 2021, date of current version November 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125000

Meta-Optimization of Bias-Variance Trade-Off in Stochastic Model Learning

TAKUMI AOTANI, (Graduate Student Member, IEEE), TAISUKE KOBAYASHI, (Member, IEEE), AND KENJI SUGIMOTO, (Member, IEEE)
Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan

Corresponding author: Takumi Aotani (aotani.takumi.ap2@is.naist.jp)

This work was supported by the Japan Science and Technology Agency (JST) Precursory Research for Embryonic Science and Technology (PRESTO), Japan, under Grant JPMJPR20C3.

ABSTRACT Model-based reinforcement learning is expected to be a method that can safely acquire the optimal policy under real-world conditions by using a stochastic dynamics model for planning. Since the stochastic dynamics model of the real world is generally unknown, a method for learning from state transition data is necessary. However, model learning suffers from the problem of bias-variance trade-off. Conventional model learning can be formulated as a minimization problem of expected loss. Failure to consider higher-order statistics for loss would lead to fatal errors in long-term model prediction. Although various methods have been proposed to explicitly handle bias and variance, this paper first formulates a new loss function, especially for sequential training of the deep neural networks. To explicitly consider the bias-variance trade-off, a new multi-objective optimization problem with the augmented weighted Tchebycheff scalarization, is proposed. In this problem, the bias-variance trade-off can be balanced by adjusting a weight hyperparameter, although its optimal value is task-dependent and unknown. We additionally propose a general-purpose and efficient meta-optimization method for hyperparameter(s). According to the validation result on each epoch, the proposed meta-optimization can adjust the hyperparameter(s) towards the preferred solution simultaneously with model learning. In our case, the proposed meta-optimization enables the bias-variance trade-off to be balanced for maximizing the long-term prediction ability. Actually, the proposed method was applied to two simulation environments with uncertainty, and the numerical results showed that the well-balanced bias and variance of the stochastic model suitable for the long-term prediction can be achieved.

INDEX TERMS Machine learning algorithms, systems modeling, Pareto optimization, bias-variance trade-off.

I. INTRODUCTION

Reinforcement learning (RL) [1] is one of the promising methods for robots to adaptively acquire their own policies in the real world. In recent years, RL has been applied in environments with high uncertainty, where there are multiple actors (eg. human-containing systems [2], [3] and multi-agent systems [4], [5]). RL-based agents attempt stochastic actions during exploration, which may have a negative impact on the environment. Safe learning, which mitigates risk during exploration such as collision, is required in these environments.

Model-based RL is expected to take safety into account by using stochastic dynamics models in planning. One such approach is model predictive shielding (MPS), which utilizes an idea called *shielding* [6], [7]. MPS uses the stochastic dynamics model, and *shielding* intervenes in the agent's action to ensure that state transitions satisfy safety constraints. Intervention by *shielding* is triggered when the agent is prone to go into the states outside of the safety constraints (predicted by the dynamics model). Similarly, tube model predictive control (tube MPC) is another model-based planning method that explicitly considers safety. Uncertainty in dynamics prediction propagates in a time-evolving manner. The region surrounding a possible transition state is called *tube*, and planning within this *tube* is performed in tube MPC [8]–[10]. However, these methods do not provide how

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Alatas.

to obtain the accurate dynamics model for the target environment, while that is mandatory in them.

Learning with the deep neural networks (DNNs) is widely applied to achieve the stochastic model prediction in recent years. The objective function for training is generally formulated as the minimization problem of expected prediction loss for the next state. The expectation is regarded as a first-order moment, namely, this approach does not optimize higher-order moments, such as variance. Hence, if a large prediction error occurs even once during the prediction of a long-term trajectory, all subsequent states will become outliers. With the fact that the prediction model can be given as a probability distribution, Bayesian theory has been appropriately utilized to consider the uncertainty of the model [11], [12]. In particular, Chua *et al.* [12] has proven a simple ensemble method, in which multiple models are prepared and trained simultaneously. In their work for the latest model-based RL, the learned models made stable planning possible. The models are however approximated with DNNs, hence the number of parameters would be huge if multiple models are used. In addition, the number of models required for the target environment must be determined empirically. Although robust control theory that explicitly considers model uncertainty or input uncertainty have been proposed [13], [14], they assume linearity and cannot be directly applied to nonlinear stochastic models. A learning method that predicts the stochastic nonlinear dynamics by DNNs with limited size is required for practical use.

If DNNs with limited size try to reduce the uncertainty of model, a well-known problem in regression, called the bias-variance trade-off [15], cannot be ignored. While the bias can be reduced by the conventional minimization problem of the expected loss, that raises the risk degrading the generalization performance caused by the increase of the variance. On the other hand, if the variance is somehow reduced excessively, the average prediction performance would be deteriorated. Even though both bias and variance can be reduced in DNNs with sufficiently large size [16], [17], the bias-variance trade-off still need to be considered for practical use. We notice the important fact that the optimal balance of the trade-off is task-dependent and basically non-trivial. With the same awareness of the issue, various bias-variance decompositions for regression problems have been proposed [18]–[21]. The conventional methods, however, cannot be applied to model training with DNNs for model-based RL. Therefore, a new decomposition suitable for sequential training of the deep neural networks is needed.

In this paper, we propose a comprehensive algorithm to obtain an optimal balance between bias and variance for the meta-objective required in model-based RL. We first attempt to formulate the bias-variance trade-off as a multi-objective optimization (MOO) problem. From a statistical point of view on the loss of the entire dataset, we note that the bias and variance can be represented by the mean loss and the worst loss. The argument begins with the fact that the expected loss function of the minimization problem for a given dataset is an equally-weighted sum of the losses for each data. In other words, the conventional method can be interpreted as a method to obtain a Pareto solution by evaluating the loss of each data equivalently. The optimization based on scalarization with the linear weighted sum, however, cannot obtain Pareto solutions on the non-convex part. Therefore, we apply the augmented weighted Tchebycheff scalarization [22], [23], which can effectively find non-convex Pareto frontiers, to each data loss. The weighted sum of the mean loss and the worst loss is derived as a new minimization target in this scalarization. That is, the next step is to apply the augmented weighted Tchebycheff scalarization again so that arbitrary Pareto solutions among the statistics (i.e. mean and worst) can be found.

The balance between the bias and the variance can be adjusted using a hyperparameter given by the above process. The Pareto solution to be used, called the preferred solution, is therefore selected by tuning this hyperparameter from the set of Pareto solutions according to the higher-level objective in general. The simplest way to find a preferred solution is brute-force exploration of the related hyperparameter(s), although this approach is computationally expensive as a matter of course. As a more advanced method, meta-optimization of parameters included in lower-level objectives [24] has been developed with several forms: e.g. gradient descent (GD) [25]–[28]; RL [29]–[31]; evolutionary search (ES) [32]–[34]; and Bayesian optimization (BO) [35], [36]. However, these conventional methods have the limitation of assuming the differentiability of the meta objective and/or requiring multiple lower-level learning trials.

Hence, we propose a general-purpose and efficient meta-optimization method based on a policy gradient method [37]. Specifically, the proposed method learns a policy that outputs hyperparameters stochastically. In each epoch, twin models are trained using the mean and sampled values of the policy, respectively, and the trained models are validated against the meta objective. The difference in the validation results would be related to only the sampled hyperparameters, not the training results, and therefore, the log-likelihood of the policy with the sampled hyperparameters, weighted by the difference in the validation results, can be maximized so as to optimize the meta objective. Before starting the next epoch, the twin models are remade from either of the old twins. In this method, the meta objective is not differentiated, and multiple trials are not necessary since the hyperparameters are optimized at the same time as the DNNs parameters.

The contributions in this paper are three folds:

1) Formulation of the bias-variance trade-off as a MOO problem
2) Development of a general-purpose and efficient meta-optimization method
3) Numerical verification of the proposed formulation with the meta-optimization on two simulations for the

environments with uncertainty due to human operation and presence of other agents

II. RELATED WORK

A. BIAS-VARIANCE DECOMPOSITION

Traditionally, the problem of bias-variance trade-off has been pointed out in data-driven learning. Various bias-variance decompositions are presented for several loss functions (e.g. mean-squared loss [15], [38], zero-one loss [38], and log-likelihood type loss [39]) used in regression. For the selection of regression models to avoid overfitting, the bias-variance have been decomposed as accuracy and complexity according to the information criterion [40]. In recent years, several bias-variance decompositions have been proposed to treat the trade-off as a MOO problem. In this section, we characterize the proposed decomposition method by comparing it with conventional methods.

A semi-parametric Gaussian copula regression that is robust to multiple datasets is proposed [20]. In generating the cumulative distribution function used for the prior distribution, the parameters of the quantile estimate adjust the bias and variance. However, the idea is not directly applicable to model training with neural networks.

A decomposition method has been proposed for model selection of Bayesian networks, where the evaluation function is defined by the accuracy and complexity using the minimal description length (MDL) [18]. Applying MDL to general DNNs where each node (neuron) has a real number of outputs and is large in scale, is, however, difficult. Since the data is sampled online, selecting the best model in advance, is also not suitable for the model-based RL.

Several methods have been presented for RL, focusing on the bias-variance trade-off of the policy gradient estimation. The method of using regularization by a Kullback–Leibler divergence for variance reduction [19] is discussed by restricting the problem to hyperparameters used in RL. A method that deals with the merge of gradients appearing in off-policy and on-policy learning [21] is also a decomposition method unique to RL.

Although various bias-variance decompositions have been proposed as described above, the methods suitable for safe model learning used in model-based RL, which is the target of this method, have not been well investigated. The proposed method does not analyze existing loss functions such as [15], [38], [39], but defines a new loss function by providing a decomposition that deals with the bias and variance of the loss values themselves. By focusing on DNNs, which have been traditionally used for learning stochastic models of dynamics, the loss function is defined in a form that is easy to handle in model-based RL. Furthermore, the proposed loss function is naturally derived by interpreting the conventional loss function as a MOO problem, and is not applied at the model selection stage as in [18]. This feature is an essential condition for model-based RL, which assumes online learning.

B. META-OPTIMIZATION

A learning algorithm trains DNNs based on a task-specific (low-level) loss function. According to a user-desired (high-level) meta-objective (e.g. generalizing across different tasks and long-term prediction accuracy like our setting), meta-optimization methods aim to optimize hyperparameters in the learning algorithm and/or the low-level loss function. The reason why various methods have been proposed is that the conditions to be satisfied differ depending on the problem. In this section, we qualitatively check the performance of the conventional and proposed meta-optimization methods while summarizing the necessary requirements for general meta-optimization. The comparison results are summarized in Table 1.

First, minimization of the low-level loss function is generally with high computational cost due to large dataset for training DNNs. The meta-optimization methods should be, therefore, highly efficient. The number of hyperparmeters to be optimized is problem-dependent (e.g. one in our case and hundreds in optimization of the architecture of DNNs), and therefore, scalability is important. Furthermore, versatility is also important to employ arbitrary meta-objective, loss function, architecture of DNNs, and so on. In particular, differentiability of meta-objective function over hyperparameters cannot be assumed since it absolutely limits the applicable problems. Hence, the following four requirements are raised: i) *high efficiency*; ii) *high scalability*; iii) *arbitrariness of target*; and iv) *no use of gradient*.

a: HIGH EFFICIENCY

Since meta-optimization is performed at a higher layer of the low-level learning, using the results of low-level learning is generally necessary [24]. While meta-optimization may aim to improve the efficiency of low-level learning, improving the efficiency of meta-optimization itself is also important, in order to reduce time cost and computational resources. In this evaluation, we examine whether or not to complete meta-optimization is possible from a single trial of a limited number of low-level learners.

GD-based methods [25]–[28] directly optimize the target hyperparameters, and thus have higher efficiency. RL-based methods are fundamentally less efficient because they require wide exploration and many trials [29]. Some methods [30], [31], however, achieved high efficiency with using the gradient of meta-objective or low-level loss, by limiting the application to RL. BO can also achieved high efficiency by finding the points to be explored. On the other hand, ES-based methods [32]–[34] are well known as less efficient because they use multiple trials or many low-level learners.

b: HIGH SCALABILITY

When meta-optimization methods are applied to optimize a large number of hyperparameters, the complexity of search space increases combinatorially. Since the order of

TABLE 1. Comparison of recent meta-optimization methods with four indices: only our method satisfies all the indices introduced here.

Approach	Method	High efficiency	High scalability	Arbitrariness of target	No use of gradient
	Our method	✓	✓	✓	✓
Gradient descent	[25]–[27]	✓	✓		
	[28]	✓	✓	✓	
Reinforcement learning	[29]		✓	✓	✓
	[30], [31]	✓	✓		
Evolutionary search	[32]–[34]			✓	✓
Bayesian optimization	[35], [36]	✓		✓	✓

computational complexity with respect to the number of hyperparameters is directly related to the versatility, a scalable method is desired to be developed.

Methods that aim for local optima based on direct information, such as hyperparameter-dependent gradients, generally have high scalability [25]–[31] by not using global information in the search space. On the other hand, in heuristic methods [32]–[34], it is intractable to find the optimal solution without many search points on large search space. The convergence is sacrificed in exchange for not limiting the search space, which is the reason for the reduced scalability. In addition, BO [35], [36] uses the Gaussian process [41] to estimate the model, the computational cost explodes with respect to the number of search points because samples of various values need to obtain the global shape of the objective function.

c: ARBITRARINESS OF TARGET

When dealing with MOO problems such as the bias-variance trade-off, the meta-objective for selecting one of the Pareto solution sets cannot be assumed in advance.. A method that can handle arbitrary meta-objective, rather than a method requiring the specific meta-objective format, is essential.

Many methods specialize in the typical meta-objectives: domain generalization [25], [26]; surrogate loss [27]; RL [30], [31], [33]; winning in games [34]; and low-level learning loss itself [32]. The meta-heuristics used in ES-based methods (e.g. CMA-ES [42] and evolution strategies [43]), however, can potentially be extended to arbitrary targets. BO [35], [36] is also suitable for handling the arbitrary targets due to its statistically-generalized design. Some methods [28], [29] have been proposed that can also handle a relatively wide range of meta-objectives, although they still restrict the format of the meta-objectives and the information required.

d: NO USE OF GRADIENT

A meta-objective for extracting a preferred solution to a MOO problem may be given only an evaluation value, and differentiability in the hyperparameters of interest cannot be assumed. This metric is marked whether the gradient with the target hyperparameter is used for meta-optimization.

In GD-based methods [25]–[28], the use of gradient is the key to meta-optimization. One method [29] based on RL, however, avoids the differentiation of meta-objective by using stochastic policy. ES-based methods [32]–[34] and BO [35], [36] are sampling-based and do not require gradient information.

e: PROPOSAL

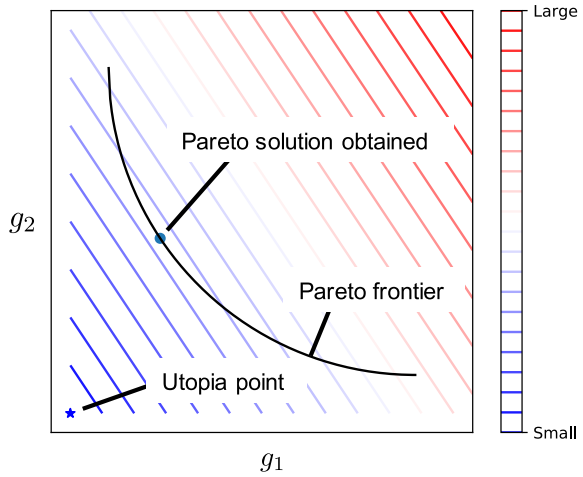
The proposed method performs meta-optimization simultaneously with low-level learning, and the low-level learning is limited to a single trial (but with two learners). Both the conventional method [29] and the proposed method use the stochastic policy for meta-optimization. However, the proposed method avoids state-dependent exploration by using only a policy-gradient method instead of RL. Another advantage of using only the policy-gradient method is that there is no need to design the state on which the meta-objective depends. Two ideas provide these advantages. The first is that the proposed method identifies the local gradient direction from the difference in evaluation between the baseline and the sample values. The other is to match the states of the two low-level learners at the beginning of each epoch, thus eliminating the need to take the states into account for the difference in learning results. In addition, the meta-objective only needs to be given a numerical scalar value as an evaluation of the low-level learners, and there is no need to assume either type or differentiability.

III. PRELIMINARIES

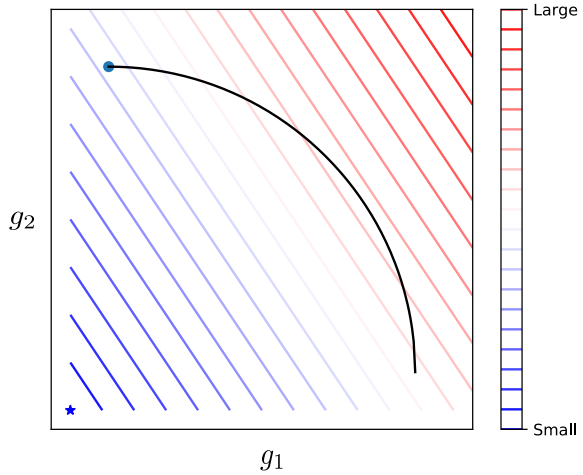
A. STOCHASTIC MODEL LEARNING IN MARKOV DECISION PROCESSES

The stochastic dynamics of the environment including the agent is formulated by a Markov decision process (MDP) in RL. Given a state $s_t \in \mathcal{S} \subset \mathbb{R}^{d_s}$ and an action $a_t \in \mathcal{A} \subset \mathbb{R}^{d_a}$ at time t, the next state s_{t+1} is assumed to be stochastically sampled from the environment-specific state transition probability $p_e(s_{t+1} \mid s_t, a_t)$. Here, d_s and d_a are the dimension sizes of the state and the action, respectively. p_e is, however, generally unknown, and model-based RL approximates it to be a model constructed DNNs parameterized by θ, $p_m(s_{t+1} \mid s_t, a_t; \theta)$. If p_m is accurately acquired, the agent can predict the future states according to the performed actions, hence, can plan the best actions to maximize rewards from the environment.

Therefore, the goal of a stochastic model learning is usually to fit p_m to p_e through minimization of expectation of negative log-likelihood, $\ln p_m$, w.r.t. p_e. Since p_e is a black-box, the expectation is replaced by Monte Carlo method

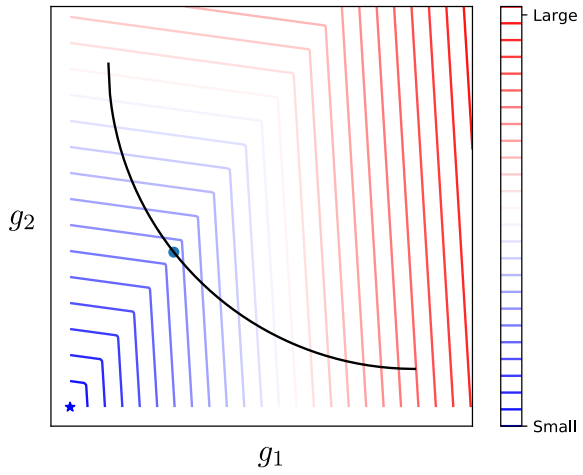


(a) Convex Pareto frontier

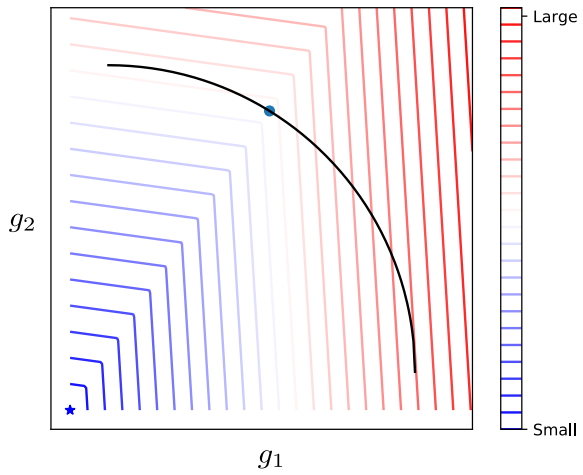


(b) Non-convex Pareto frontier

FIGURE 1. Illustration of the contour and the Pareto solution obtained by the linearly weighted sum: the shape of the contour line is linear; (a) the Pareto solution of the convex part of the Pareto frontier can be obtained; (b) however, the Pareto solution of the non-convex part of the Pareto frontier cannot be obtained.



(a) Convex Pareto frontier



(b) Non-convex Pareto frontier

FIGURE 2. Illustration of the contour and the Pareto solution obtained by the augmented weighted Tchebycheff scalarization: The shape of the contour line is a linear combination of the L-shaped line and the linear line; (a) the Pareto solution of the convex part of the Pareto frontier can be obtained; (b) and, the Pareto solution of the non-convex part of the Pareto frontier can also be obtained.

as sample mean over the dataset obtained from p_e, $D = \{(s_n,\ a_n),\ s_{n+1}\}_{n=1}^{N}$, with N tuples. More specifically, θ is optimized toward θ^*, to minimize the following formula.

$$\theta^* = \arg\min_{\theta} \mathcal{L}_i(\theta),\ \forall i \in \mathbb{N}$$
$$\mathcal{L}_i(\theta) = \frac{1}{N_i} \sum_{s_i, a_i, s_{i+1} \in D_i} -\ln p_m(s_{i+1} \mid s_i,\ a_i;\ \theta) \quad (1)$$

where $D_i \subset D$ denotes i-th mini-batch with batch size N_i extracted from D.

B. AUGMENTED WEIGHTED TCHEBYCHEFF SCALARIZATION

When considering the minimization of M objective functions $g_1, \ldots, g_M$ as a MOO problem, the optimality of the solution is defined by dominance. The solution x that satisfies the following formula dominates the solution x' and is expressed as $x \prec x'$.

$$\forall m, g_m(x) \leq g_m(x') \wedge \exists m, g_m(x) < g_m(x') \quad (2)$$

The solution that is not dominated by all other solutions is called the Pareto solution. The set of Pareto solutions is called the Pareto frontier.

The goal of MOO is to find the Pareto solution or the Pareto frontier while taking into account the trade-offs among the objective functions. To this end, in most cases, a scalarization function $h : \mathbb{R}^M \to \mathbb{R}$ with a weight vector $\boldsymbol{w} \in \mathbb{R}^M$ makes the objective function vector $\boldsymbol{g} = [g_1, \ldots, g_M]^\top$ scalar in order to transform a MOO problem into a set of single-objective optimization problems. The simplest scalarization function is the linear weighted sum in the following equation.

$$h(x) = \sum_{m=1}^{M} w_m g_m(x) \quad (3)$$

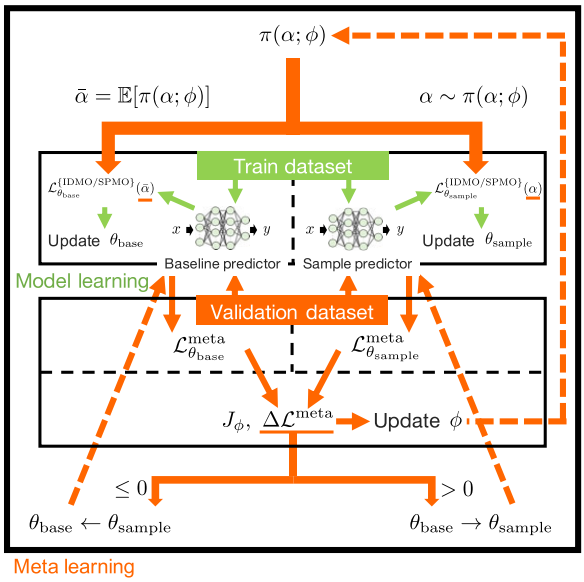


FIGURE 3. Schematic of the proposed method: Two stochastic models are learned using the training dataset, and based on the differences between them, meta policy is simultaneously optimized under the meta objective using the validation dataset.

In the case of scalarization by linear weighted sum, the contour line in the search space is just a linear line. Therefore, the Pareto solution in the non-convex part of the Pareto frontier cannot be obtained (see Figs. 1 (a), (b)).

The augmented weighted Tchebycheff scalarization, defined by the following equation, is widely used as one of the scalarization functions that can deal with non-convex Pareto frontiers [22], [44].

$$h(x) = \max_{1\leq m\leq M} w_m(g_m(x) - u_m) + \alpha \sum_{m=1}^{M} w_m(g_m(x) - u_m) \quad (4)$$

where u_m is called a utopia point that strictly dominates g_m. The contour line is a linear combination of the L-shaped line from the first term and the linear line from the second term with the hyperparameter $\alpha > 0$ (see Figs. 2 (a), (b)). As for the choice of α, the effects are noted that too small α would cause a weak Pareto solution because the effect of the second term is relatively insignificant, and too large α would make the non-convex solutions unreachable because the effect of the first term is weakened [23], [45].

IV. META-OPTIMIZATION OF BIAS-VARIANCE TRADE-OFF

A. OVERVIEW

The state transition model with the parameter θ is optimized to minimize the loss function, i.e. the expected value of the negative log-likelihood, defined in eq. (1). Although this approach is effective when the stochastic behavior is relatively small, it does not take into account the variance of losses and worst-case scenarios, which are represented by higher-order moments, and thus can lead to large prediction errors in reality. Particularly in the case of RL and MPC applications, the resulting model is used for long-term prediction, and if the prediction fails even once during the period, subsequent predictions from that point may fail. This problem is caused mainly by a bias-variance trade-off.

This paper proposes a method to adjust the balance of the bias-variance trade-off by simultaneously minimizing the expected loss and the worst loss, which are theoretically derived later. In this case, since learning the state transition model becomes a MOO problem, a loss function scalarized by the augmented weighted Tchebycheff scalarization can be applied. In light of the fact that the size of the Pareto solution set is generally innumerable, a general-purpose meta-optimization method is also proposed to obtain the preferred solution depending on the given meta-objective. A schematic diagram of the entire proposed method is shown in Fig. 3.

B. FORMULATION OF MOO PROBLEM

1) INTER-DATA MOO: IDMO

To avoid complication, the loss function of the conventional method, eq. (1), is redefined as follows:

$$\mathcal{L}_\theta^{\text{mean}} = \frac{1}{N}\sum_{i=1}^{N} l_{i,\theta} \quad (5)$$

$$l_{i,\theta} = -\ln p_m(s_{i+1} \mid s_i,\ a_i;\ \theta) - u \quad (6)$$

where u is a utopia point, which is given commonly to all data as $u = \min -\ln p_m(s_{i+1} \mid s_i,\ a_i;\ \theta)$. The above objective function can be interpreted as the linear weighted sum with objectives for each data and equivalent weights $w_i = 1/N$. That is, the conventional way can be regarded as an inter-data multi-objective (IDMO) optimization problem.

According to this interpretation, we formulate this IDMO optimization problem based on the augmented weighted Tchebycheff scalarization that can obtain all Pareto solutions.

$$\theta^* = \arg\min_\theta \mathcal{L}_\theta^{\text{IDMO}}$$

$$\mathcal{L}_\theta^{\text{IDMO}} = \frac{\tilde{\alpha}}{N}\sum_{i=1}^{N} l_{i,\theta} + \frac{1}{N}\max_i l_{i,\theta} \propto \mathcal{L}_\theta^{\text{mean}} + \alpha\mathcal{L}_\theta^{\text{worst}} \quad (7)$$

where $\mathcal{L}_\theta^{\text{worst}} = \max_i l_{i,\theta}$ is the worst loss, and $\tilde{\alpha}$ is the hyperparameter in the augmented weighted Tchebycheff scalarization. Since the solution is not changed by constant multiplication of the loss function, eq. (7) can be used under $\alpha = 1/(\tilde{\alpha}N)$.

As shown in the above formula, the application of the augmented weighted Tchebycheff scalarization to the IDMO optimization naturally leads to a loss function that explicitly considers the mean loss and the worst loss, which correspond to the bias and the variance, respectively. Therefore, the bias-variance trade-off can be adjusted by setting α appropriately.

TABLE 2. Reachability to non-convex solutions.

Loss	Non-convex solutions	
	Inter data	Statistics perspective
Vanilla loss $\mathcal{L}_\theta^{\text{mean}}$		
Proposed loss $\mathcal{L}_\theta^{\text{IDMO}}$	✓	
Proposed loss $\mathcal{L}_\theta^{\text{SPMO}}$	✓	✓

2) STATISTICS-PERSPECTIVE MOO: SPMO

The loss $\mathcal{L}_\theta^{\text{IDMO}}$ obtained above can be again interpreted as a linear weighted sum of the statistics $\mathcal{L}_\theta^{\text{mean}}$ and $\mathcal{L}_\theta^{\text{worst}}$ weighted by the ratio $1 : \alpha$. When the Pareto frontier for $\mathcal{L}_\theta^{\text{mean}}$ and $\mathcal{L}_\theta^{\text{worst}}$ is non-convex, the Pareto solutions that cannot be obtained at the statistical level would be worth considering.

We, therefore, apply the augmented weighted Tchebycheff scalarization again to such a statistics-perspective multi-objective (SPMO) optimization problem as follows:

$$\mathcal{L}_\theta^{\text{SPMO}} = \max\left(\mathcal{L}_\theta^{\text{mean}}, \alpha\mathcal{L}_\theta^{\text{worst}}\right) + \beta\left(\mathcal{L}_\theta^{\text{mean}} + \alpha\mathcal{L}_\theta^{\text{worst}}\right) \quad (8)$$

where β denotes the hyperparameter in the augmented weighted Tchebycheff scalarization. Since each loss is non-negative, its statistics, $\mathcal{L}_\theta^{\text{mean}}$ and $\mathcal{L}_\theta^{\text{worst}}$, are also non-negative, and no utopia point is needed. Note that, in the case of $\alpha \geq 1$, the above formula is essentially equivalent to eq. (7) since the first term is always $\mathcal{L}_\theta^{\text{worst}}$.

3) SUMMARY OF PROPOSED LOSSES

The properties of the loss functions based on the proposed multi-objective optimization problems are summarized in Table 2.

First, from the perspective of IDMO, the conventional loss function, the mean loss $\mathcal{L}_\theta^{\text{mean}}$, can be interpreted as a linear weighted sum of the objectives. Since the linear weighted sum cannot yield non-convex solutions, a loss $\mathcal{L}_\theta^{\text{IDMO}}$ was formulated based on the augmented weighted Tchebycheff scalarization.

Furthermore, since $\mathcal{L}_\theta^{\text{IDMO}}$ was derived as a linear sum of the mean loss and the worst loss, a loss $\mathcal{L}_\theta^{\text{SPMO}}$ was formulated by applying the augmented weighted Tchebycheff scalarization to those statistics again. The loss $\mathcal{L}_\theta^{\text{SPMO}}$ implicitly includes $\mathcal{L}_\theta^{\text{IDMO}}$, and the non-convex solutions can be obtained even in IDMO.

C. META-OPTIMIZATION OF HYPERPARAMETER

Even though the hyperparameter α of the loss functions ($\mathcal{L}_\theta^{\text{IDMO}}$ and $\mathcal{L}_\theta^{\text{SPMO}}$) shown in the previous section contributes significantly to the bias-variance tradeoff, no clear metric has been defined to determine its value. In this section, we propose a general-purpose meta-optimization method for α under an arbitrary meta-objective, $\mathcal{L}^{\text{meta}}$, given as a high-level design metric.

This meta-optimization problem can be formulated as follows:

$$\alpha^* = \arg\min_\alpha \mathcal{L}^{\text{meta}}(\theta^*(\alpha), D^{\text{val}}) \quad s.t.\ \theta^*(\alpha) = \arg\min_\theta \mathcal{L}_\theta^{\{\text{IDMO,SPMO}\}}(\alpha, D^{\text{trn}}) \quad (9)$$

where D^{trn} and D^{val} are the training and validation datasets, respectively, and are generated such that $D^{\text{trn}} \cap D^{\text{val}} = \emptyset$, $D^{\text{trn}} \cup D^{\text{val}} = D$ are satisfied.

To solve this meta-optimization problem, we first suppose that $\alpha \in [0, 1]$ (this restricted range is for distinguishing IDMO and SPMO) is sampled from a meta-policy $\pi(\alpha;\ \phi)$ constructed as a probability distribution parameterized by ϕ. The purpose is converted to optimize ϕ to minimize $\mathcal{L}^{\text{meta}}$ stochastically.

$$\phi^* = \arg\min_\phi \mathbb{E}_{\alpha\sim\pi(\alpha;\ \phi)}[\mathcal{L}^{\text{meta}}(\theta^*(\alpha), D^{\text{val}})] \quad (10)$$

The gradient of the above objective function over ϕ can be computed following the policy gradient method.

$$\begin{aligned}\nabla_\phi \mathbb{E}_{\alpha\sim\pi(\alpha;\ \phi)}[\mathcal{L}^{\text{meta}}(\theta^*(\alpha), D^{\text{val}})] &= \mathbb{E}_{\alpha\sim\pi(\alpha;\ \phi)}[\mathcal{L}^{\text{meta}}(\theta^*(\alpha), D^{\text{val}})\nabla_\phi \ln\pi(\alpha;\ \phi)] \\ &= \mathbb{E}_{\alpha\sim\pi(\alpha;\ \phi)}[\{\mathcal{L}^{\text{meta}}(\theta^*(\alpha), D^{\text{val}}) - b\}\nabla_\phi \ln\pi(\alpha;\ \phi)]\end{aligned} \quad (11)$$

where b denotes the baseline, which is not related to α. Since the expectation of the term for b is zero, it can be added freely as long as it does not depend on α, which greatly reduces the variance of the learning results.

To design b, we provide twin models, p_m^{base} and p_m^{sample}, which are with exactly the same θ before each epoch. In each epoch, they are trained with $\bar{\alpha} = \mathbb{E}[\pi(\alpha;\ \phi)]$ and $\alpha \sim \pi(\alpha;\ \phi)$, resulting in θ_{base} and θ_{sample}, respectively. Since p_m^{base} is not involved in α, $\mathcal{L}^{\text{meta}}$ with θ_{base} can be utilized as the baseline. In addition, we can separate whether the variation in $\mathcal{L}^{\text{meta}}$ comes from α or training, and extract only the contribution of α by subtracting this baseline from $\mathcal{L}^{\text{meta}}$ with θ_{sample}.

Hence, with $\mathcal{L}^{\text{meta}}(\theta_{\text{base,sample}}, D^{\text{val}}) =: \mathcal{L}^{\text{meta}}_{\theta_{\text{base,sample}}}$, the meta-objective function for ϕ, J_ϕ, can be given as follows:

$$J_\phi = \Delta\mathcal{L}^{\text{meta}} \ln\pi(\alpha;\ \phi) \quad (12)$$

$$\Delta\mathcal{L}^{\text{meta}} = \mathcal{L}^{\text{meta}}_{\theta_{\text{sample}}} - \mathcal{L}^{\text{meta}}_{\theta_{\text{base}}} \quad (13)$$

where the expectation operation in eq. (11) is eliminated by one-sample Monte Carlo approximation as well as the standard policy gradient method.

Afterwards, to start the new epoch with the twin models parameterized by the same θ, they are renewed from the superior model.

$$\begin{cases}\theta_{\text{base}} \leftarrow \theta_{\text{sample}} & (\Delta\mathcal{L}^{\text{meta}} \leq 0) \\ \theta_{\text{sample}} \leftarrow \theta_{\text{base}} & (\Delta\mathcal{L}^{\text{meta}} > 0)\end{cases} \quad (14)$$

The proposed method can perform the model learning and the meta-optimization simultaneously at each epoch.

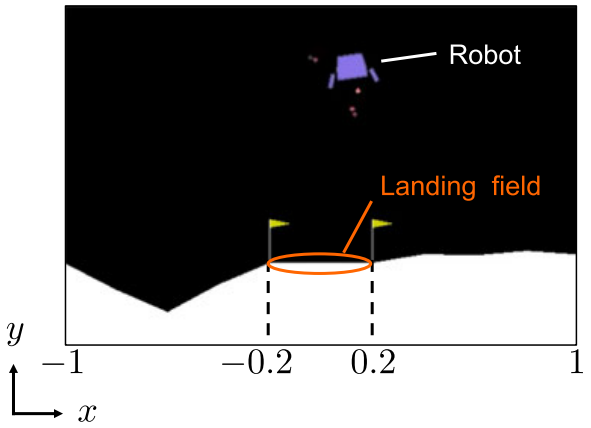


FIGURE 4. Human-operated single-agent environment: The robot is operated by an expert (human); The expert aims to land the robot on the landing field at zero speed.

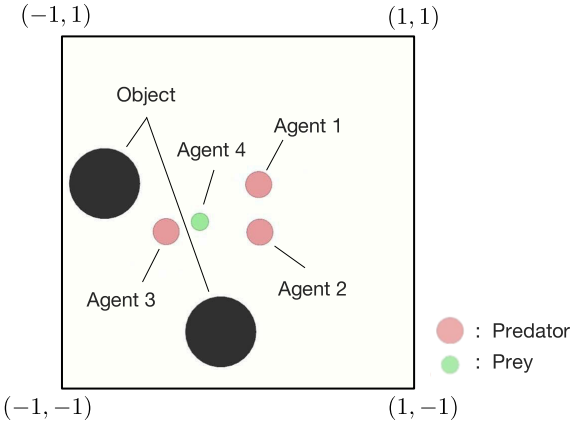


FIGURE 5. Multi-agent environment: four agents work in the same environment; the task of agents 1–3 (predators) is to catch an agent 4 (prey); the task of the agent 4 is to run away from the agent 1-3 on the screen.

In addition, the additional computational cost for the meta-optimization is only to learn the twin model, resulting in sufficiently low computational cost and high efficiency. There are no requirements on the learning algorithm, the meta-objectives, and so on. That is, the proposed method is sufficiently versatile. Even if the number of hyperparameters is increased, the computational cost of the policy gradient method is merely proportional to it, keeping high scalability.

V. EXPERIMENTS

A. COMMON CONDITIONS

In order to validate the effectiveness of the proposed method, model learning with two types of meta-objective is conducted with datasets collected in numerical simulations. One meta-objective is simply the minimization of the linear weighted sum of the mean and worst losses to verify whether the proposed meta-optimization method succeeds in making reasonable adjustments. Another is the minimization of the negative log-likelihood in the long-term prediction as more practical case. In the following, the proposed method combining IDMO/SPMO and meta-optimization will be referred to as IDMO+MO/SPMO+MO, respectively.

TABLE 3. Hyperparameters in human-operated single-agent environment.

Hyperparameter	Value
Learning rate of model predictor	0.0001
Batch size of model predictor	64
β for $\mathcal{L}_\theta^{\mathrm{SPMO}}$	0.0001
Learning rate of meta-policy	0.0001

TABLE 4. Hyperparameters in multi-agent environment.

Hyperparameter	Value
Learning rate of model predictor	0.00001
Batch size of model predictor	32
β for $\mathcal{L}_\theta^{\mathrm{SPMO}}$	0.0001
Learning rate of meta-policy	0.0001

The model to learn the stochastic dynamics is configured as a three-layered neural network with 100 neurons in each hidden layer, and the meta-policy is configured as a one-layered neural network with 100 neurons in hidden layer, in all trials. All hidden layers are configured as fully connected layers, and the activation function is the ReLU (Rectified Linear Unit). These networks output a multivariate diagonal Gaussian distribution and a Beta distribution, respectively, and implemented by Pytorch. They are optimized with one of the state-of-the-art stochastic GD optimizer, t-Adam [46], which is robust to noise and outliers in dataset.

Two types of simulation environments are prepared: 1) a human-operated single-agent environment; and 2) a multi-agent environment. Details of each environment is described below.

In both environments, their dataset were randomly divided in proportions such that $N_{\mathrm{trn}} : (N_{\mathrm{val}} + N_{\mathrm{tst}}) = 7 : 3$ and $N_{\mathrm{val}} : N_{\mathrm{tst}} = 7 : 3$, where N_{trn}, N_{val}, and N_{tst} are the numbers of training, validation, and test data, respectively. For each training condition, 10 trials of model learning in 200 epochs are performed with different random seeds to confirm the statistical performance.

1) HUMAN-OPERATED SINGLE-AGENT ENVIRONMENT

This environment is "LunarLanderContinuou-v2" provided by OpenAI Gym [47] (see Fig. 4). The robot moves in the environment by the thrust of the main engine and the left and right engines. The state space of the robot is eight-dimensional: two-dimensional absolute position and velocity; attitude and angular velocity; and two states of contact between the ground and each foot. The action space of the robot is two-dimensional: thrust of main engine; and thrust of left and right engines. The task to be accomplished is to softly land on the landing field and stop.

To train the stochastic dynamics model, we manually collected state transition data $\{(s_t,\ a_t),\ s_{t+1}\}$. The data collector

generated the action sequences as an expert with the aim of realizing the task. The number of data in this dataset D is $N = 70,394$. The manually-collected dataset would contain bias in the states visited, and such a dataset is prone to bias and/or variance of the trained model.

In this environments, all the following trials are conducted with the hyperparameters shown in Table 3.

2) MULTI-AGENT ENVIRONMENT

This environment based on [48] consists of four agents and two objects, which are randomly placed at the beginning of each episode (see Fig. 5). The source codes of this environment can be downloaded from Github: `https://github.com/openai/multiagent-particle-envs`). The state space of the agents 1–3 (predators) is 16-dimensional: two-dimensional absolute position and velocity of itself; two-dimensional relative positions of the other agents and objects; and two-dimensional velocity of the agent 4 (prey). The state space of the agent 4 (prey) is fourteen-dimensional: two-dimensional absolute position and velocity of itself; and two-dimensional relative positions of other agents. Each agent has a discrete action space for determining the moving direction (up/down/left/right).

This experiment can be regarded as a partially observable MDP (POMDP), where each agent does not share the actions of all the agents and the agent 4 does not know the positions of two objects. As a result, it is difficult to infer the true dynamics p_e from the observable states of each agent, and the model p_m will always contain uncertainty. Therefore, higher-order moments (the worst loss in our case) must be properly considered.

To make the dataset for this environment, the action of each agent at each time was designed as follows:

- Predator (i.e., the agents $i \in \{1, 2, 3\}$) pursues the agent 4.

$$a_i = \begin{cases} a_{\text{left}} & |d^{\text{h}}_{i,4}| \geq |d^{\text{v}}_{i,4}| \ \& \ d^{\text{h}}_{i,4} \leq 0 \\ a_{\text{right}} & |d^{\text{h}}_{i,4}| \geq |d^{\text{v}}_{i,4}| \ \& \ d^{\text{h}}_{i,4} > 0 \\ a_{\text{down}} & |d^{\text{h}}_{i,4}| < |d^{\text{v}}_{i,4}| \ \& \ d^{\text{v}}_{i,4} \leq 0 \\ a_{\text{up}} & |d^{\text{h}}_{i,4}| < |d^{\text{v}}_{i,4}| \ \& \ d^{\text{v}}_{i,4} > 0 \end{cases} \tag{15}$$

- Prey (i.e., the agent 4) run away from the agents 1–3.

$$a_4 = \begin{cases} a_{\text{left}} & |d^{\text{h}}_{4,i^{\text{h}}_{\min}}| \leq |d^{\text{v}}_{4,i^{\text{v}}_{\min}}| \ \& \ d^{\text{h}}_{4,i^{\text{h}}_{\min}} > 0 \\ a_{\text{right}} & |d^{\text{h}}_{4,i^{\text{h}}_{\min}}| \leq |d^{\text{v}}_{4,i^{\text{v}}_{\min}}| \ \& \ d^{\text{h}}_{4,i^{\text{h}}_{\min}} \leq 0 \\ a_{\text{down}} & |d^{\text{h}}_{4,i^{\text{h}}_{\min}}| > |d^{\text{v}}_{4,i^{\text{v}}_{\min}}| \ \& \ d^{\text{v}}_{4,i^{\text{v}}_{\min}} > 0 \\ a_{\text{up}} & |d^{\text{h}}_{4,i^{\text{h}}_{\min}}| > |d^{\text{v}}_{4,i^{\text{v}}_{\min}}| \ \& \ d^{\text{v}}_{4,i^{\text{v}}_{\min}} \leq 0 \end{cases},$$

$$i^{\text{h}}_{\min} = \arg\min_i |d^{\text{h}}_{4,i}|, \quad i^{\text{v}}_{\min} = \arg\min_i |d^{\text{v}}_{4,i}| \tag{16}$$

where $a_{\{left,right,down,up\}}$ represents left, right, down, up movement, $d^{\{\text{h,v}\}}_{i,j}$ is the relative position of the agent j in the horizontal, vertical direction from the agent i. The action of each agent is collected to keep each agent within 90% of the vertical and horizontal limits of the screen. With such ad-hoc controllers, the dataset D is collected with $N = 30,000$.

In this environment, all the following trials are conducted with the hyperparameters shown in Table 4. Note that all the agents are with the same hyperparameters, although the results of random initialization are different.

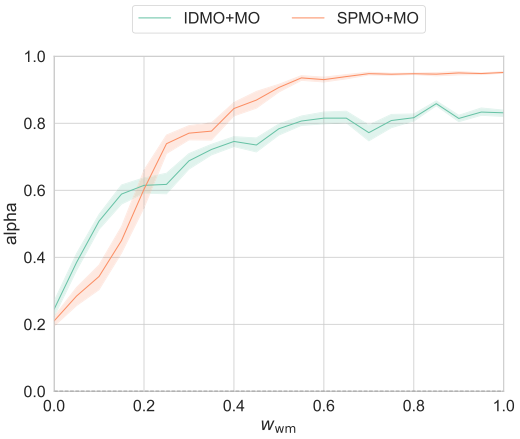


FIGURE 6. Learning results of α on human-operated single-agent environment and eq. (17): as w_{wm} increased, the α approached 1, and vice versa; the positive correlation between w_{wm} and α was captured.

B. META-OBJECTIVE: LINEAR WEIGHTED SUM OF MEAN AND WORST LOSSES

In this validation, the meta objective is defined as the linear weighted sum of the mean and worst losses for the validation dataset D^{val}, as shown in the following equation.

$$\mathcal{L}^{\text{meta}}_\theta = (1 - w_{\text{wm}})\mathcal{L}^{\text{mean}}_\theta(D^{\text{val}}) + w_{\text{wm}}\mathcal{L}^{\text{worst}}_\theta(D^{\text{val}}) \tag{17}$$

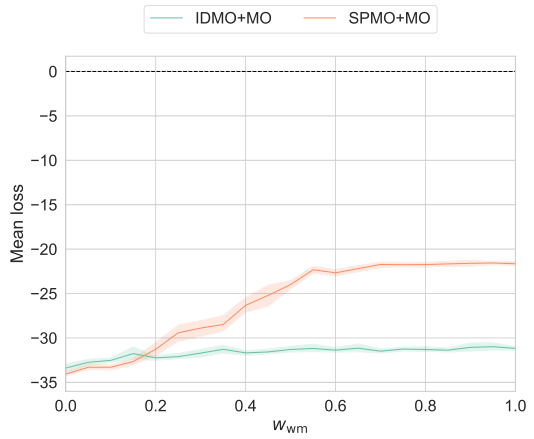
where $w_{\text{wm}} \in [0, 1]$ denotes the priority of the worst loss. Bias and variance must be adjusted for optimization of this meta-objective. The proposed method is applied to each of the 21 loss functions generated by varying w_{wm} with 0.05 increments in the range $[0, 1]$, thereby validating the meta-optimization for α, which should be positively correlated with (but not linearly proportional to) w_{wm}.

1) HUMAN-OPERATED SINGLE-AGENT ENVIRONMENT

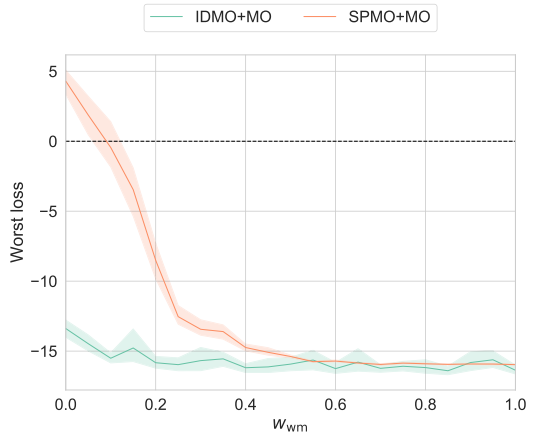
The learning results by the proposed method in human-operated single-agent environment are shown below. Each value in the following graphs represents the mean and 95 % confidence interval over 10 trials of the values obtained in the final five epochs.

The transition of the learned α with evenly-spaced w_{wm} is shown in Fig. 6. As can be seen in Fig. 6, α tends to increase with the increase of w_{wm} for both IDMO+MO and SPMO+MO methods. In other words, the proposed meta-optimization was able to capture the positive correlation between the two variables α and w_{wm}.

The transitions of mean and worst losses shown in Figs. 7 (a) and (b), show that the learning results converge to different values depending on the change of α.



(a) Mean loss



(b) Worst loss

FIGURE 7. Learning results of the mean and worst losses on human-operated single-agent environment and eq. (17): (a) the mean loss was decreased as w_{wm} increased; (b) the worst loss was increased as w_{wm} increased; as a result, a model suitable for the meta-objective was learned.

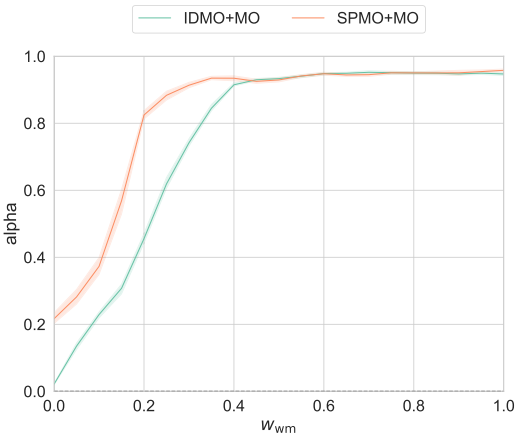


FIGURE 8. Learning results of α on multi-agent environment and eq. (17): as w_{wm} increased, the α approached 1, and vice versa; the positive correlation between w_{wm} and α was captured.

Since minimizing the mean loss of the validation data is close to the meta-objective with small w_{wm}, the stochastic model was trained to focus on the mean loss at the expense of the worst loss, and vice versa. Compared to SPMO+MO, IDMO+MO shows less variation in the learning results, even though α varies in a similar range. This is because SPMO can cover all Pareto solutions in $\alpha \in [0, 1]$, whereas IDMO is affected by the mean loss even in $\alpha = 1$. Probably thanks to this capability of SPMO, it succeeded in minimizing the mean loss with small w_{wm} less than one of IDMO, and the worst loss with large w_{wm} more stably than IDMO (i.e. smaller confidence interval).

A simple simulation with the meta-objective of minimizing the weighted sum of the mean and worst losses for the validation data shows numerically that the proposed meta-optimization method can adaptively adjust the bias-variance trade-off at the same as model learning. Since the simulation environment is human-operated single-agent environment, the proposed method would be effective in dealing with uncertainty in human-involved biased datasets. In particular, the comparison results show that SPMO+MO can handle a wider range of trade-off than IDMO+MO.

2) MULTI-AGENT ENVIRONMENT

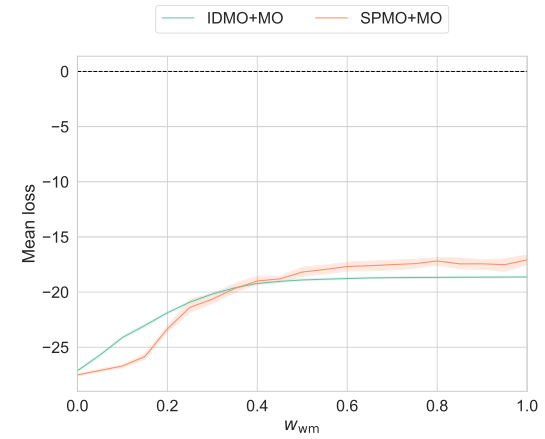
The results of applying the proposed method into the multi-agent environment are shown below, as well in the previous section.

The transition of the learned α with evenly-spaced w_{wm} is shown in Fig. 6. As well in the human-operated single-agent environment, the proposed meta-optimization method was able to captured the positive correlation between α and w_{wm} in both model learning methods. However, α saturated near 1 even when w_{wm} was relatively small compared to Fig. 6, suggesting that this environment is prone to large variance.

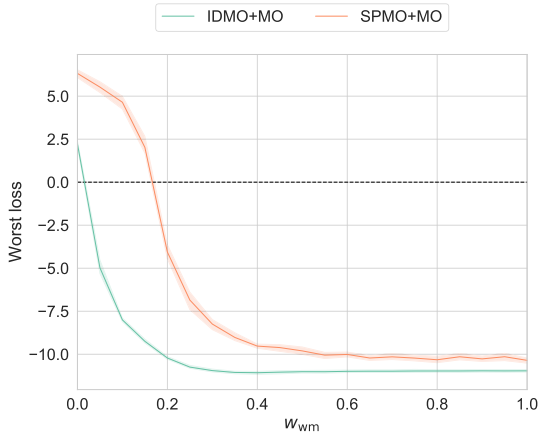
The mean and worst losses of the learning results (see Fig. 9) also show the similar tendency to Fig. 7, although the range of change in the case with IDMO was increased since the variance would be dominant. In addition, it can be seen that the model accuracy of SPMO was inferior to that of IDMO in both mean and worst losses when prioritizing the worst loss with large w_{wm}. This is probably because IDMO always uses the gradients of both the mean and worst losses, while SPMO uses either of them per batch depending on the max operator. Therefore, even with the same epochs, the number of uses of data to update the parameters is reduced, resulting in delaying the model learning itself. This drawback may be mitigated by annealing β from the large initial value, for example.

C. META-OBJECTIVE: ACCURACY OF LONG-TERM PREDICTION

In this validation, the meta-objective is defined as the mean loss in long-term prediction for the validation dataset D^{val},



(a) Mean loss



(b) Worst loss

FIGURE 9. Learning results of the mean and worst losses on multi-agent environment and eq. (17): (a) the mean loss was decreased as w_{wm} increased; (b) the worst loss was increased as w_{wm} increased; as a result, a model suitable for the meta-objective was learned.

as shown in the following equation.

$$\begin{aligned}
\mathcal{L}_\theta^{\mathrm{meta}} &= \frac{1}{K}\sum_{k=1}^{K} l_\theta^{(H,k)} \\
l_\theta^{(h,k)} &= -\ln p_m(s_{t_k+h+1} \mid \bar{s}_{t_k+h},\ a_{t_k+h};\ \theta) - u \\
\bar{s}_{t_k+h} &= \begin{cases} s_{t_k} & h = 0 \\ \mathbb{E}[p_m(\cdot \mid \bar{s}_{t_k+h-1},\ a_{t_k+h-1};\ \theta)] & \text{otherwise} \end{cases}
\end{aligned} \tag{18}$$

where H denotes the horizon of the prediction period and K denotes the number of sequences. Namely, it predicts the state sequence based on the state at t and the action sequence from t in the dataset, and the prediction accuracy at the end of the prediction period is employed as the meta-objective.

In such a long-term prediction, the stochastic model is desired to be trained not only to improve the accuracy of the one-step prediction, but also to avoid outliers during the prediction period. This meta-objective, therefore, requires the optimal balance of the bias-variance trade-off, which is difficult to be revealed analytically. In order to verify whether the proposed meta-optimization method on SPMO can properly find the optimal balance, we experiment with $H = \{1, 10\}$ as below.

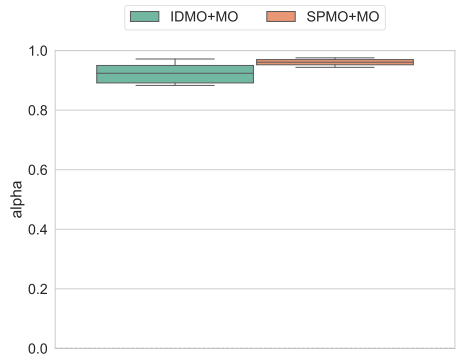


FIGURE 10. Learning results of α on human-operated single-agent environment and eq. (18) with $H = 10$: α was learned to be close to 1; the worst loss was emphasized in the long-term prediction in this environment.

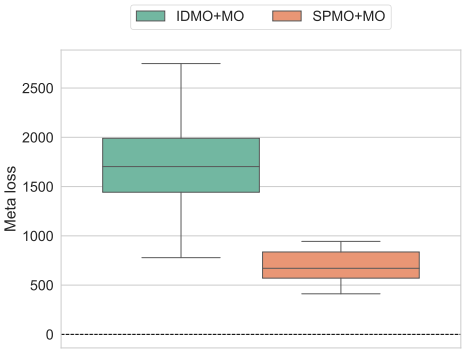


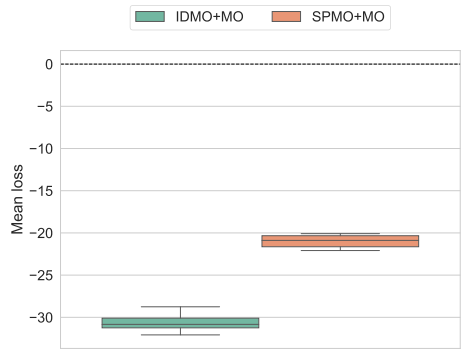
FIGURE 11. The results of the scores on human-operated single-agent environment and eq. (18) with $H = 10$: the score was reduced more in the case of SPMO+MO than in the case of IDMO+MO; namely, SPMO+MO improved the meta-objective.

1) HUMAN-OPERATED SINGLE-AGENT ENVIRONMENT

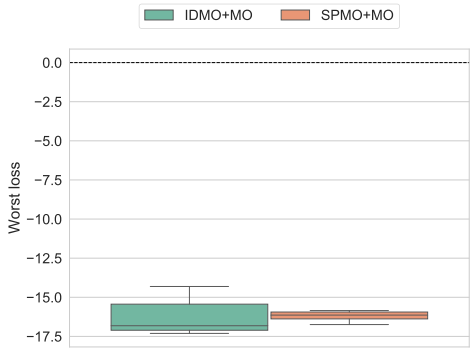
The learning results as box plots for the values obtained in the final five epochs in the human-operated single-agent environment are shown below.

First, the results for the time-horizon $H = 10$ are shown. Fig. 10 shows the results for meta-learned α. In both the SPMO+MO and IDMO+MO cases, α converged to around 1, indicating that the meta-optimization was done in a way that emphasized the worst loss.

The meta-objective obtained by the baseline model is shown in Fig. 11. Fig. 11 shows that smaller meta-objective is obtained by using SPMO+MO. The reason for this can be understood from the results of the mean and worst losses for each method shown in Figs. 12 (a) and (b). In the case of IDMO+MO, the mean loss was still minimized even under $\alpha \simeq 1$, and the gap between it and the worst loss was enlarged. This gap would cause overlearning to the mean loss, and induced outliers during the long-term prediction. In contrast, SPMO+MO obtained the larger mean loss than that of



(a) Mean loss



(b) Worst loss

FIGURE 12. Learning results of the mean and worst losses on human-operated single-agent environment and eq. (18) with $H = 10$: IDMO+MO obtained the large difference between the mean and worst losses, which means that the variance of the learned stochastic model was large; in contrast, SPMO+MO succeeded in keeping the difference between the mean and worst losses small, resulting in that fatal errors in long-term prediction were less likely to occur, as indicated in Fig. 11.

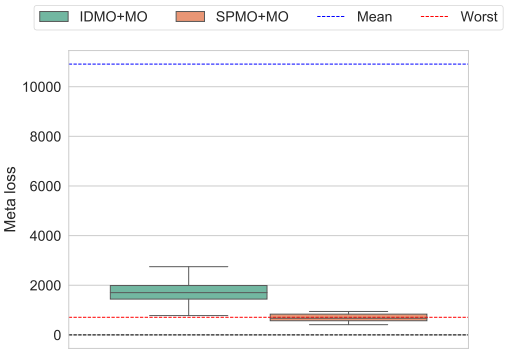


FIGURE 13. The comparison of the scores on human-operated single-agent environment and eq. (18) with $H = 10$: the scores of both SPMO+MO and IDMO+MO were reduced more in the cases of the maen loss, and only SPMO yielded results comparable to the case of the worst loss; namely, the proposed methods leads to a better Pareto solution than the conventional method, and the results suggest that convergence to the worst loss was the optimal solution.

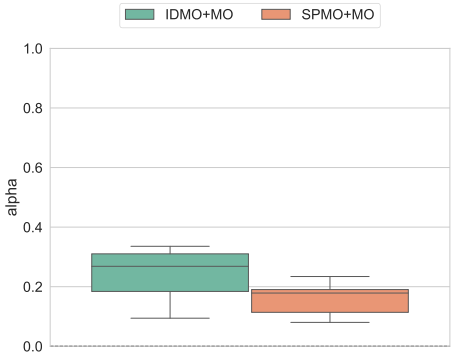


FIGURE 14. Learning results of α on human-operated single-agent environment and eq. (18) with $H = 1$: α was adjusted to be close to 0; the mean loss was emphasized to predict only the next step ($H = 1$ with the validation dataset.

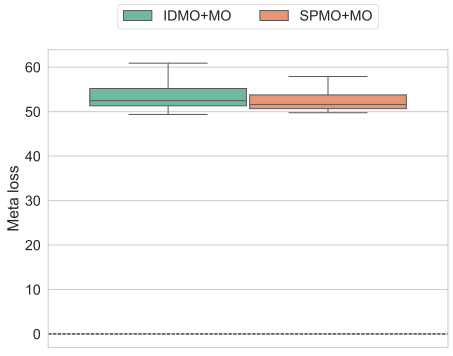
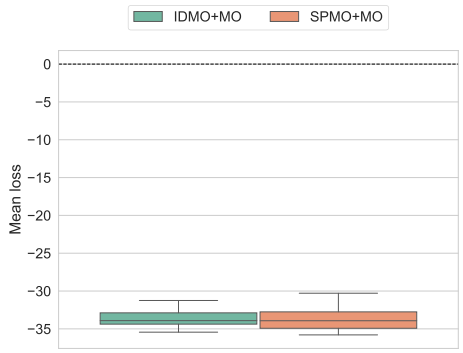


FIGURE 15. The results of the scores on human-operated single-agent environment and eq. (18) with $H = 1$: The scores obtained from the meta-optimization were comparable for both methods.

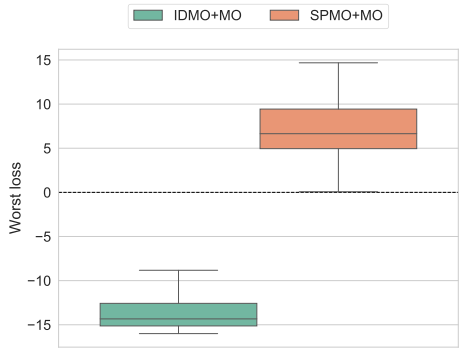
IDMO+MO, but the gap between it and the worst loss and the variance of their losses were smaller, thereby achieving the stable prediction without overlearning. In addition, Fig. 13 shows that, both proposed methods reduce the meta-objective more than the conventional method using mean loss (Mean). Only SPMO+MO resulted in the same degree of reduction as the case of using the worst loss (Worst).

Next, the results for the time-horizon $H = 1$ are described below. According to the minimization results for α shown in Fig. 14, the smaller α, i.e. prioritizing the mean loss, would be better for this setting. This is a reasonable result because the meta-objective is the same as the low-level loss for model training when $\alpha = 0$, namely the worst loss is no longer considered. SPMO+MO succeeded in obtaining the smaller α than one of IDMO+MO, which may yield a slightly better result in the meta-objective shown in Fig. 15.

It is clear from Fig. 16, which shows the mean and worst losses with $H = 1$, and Fig. 12 why α was not sufficiently small in IDMO+MO. That is, although IDMO+MO obtained the different α, the mean and worst losses were almost the same, indicating a low dependency on α. On the other hand, in SPMO+MO, the mean loss was minimized to the same level as in IDMO+MO, and the worst loss was explicitly ignored instead. Such a high dependency on α enables SPMO+MO to find the preferred solution that satisfies the meta-objective as much as possible. Fig. 17 shows that, both



(a) Mean loss



(b) Worst loss

FIGURE 16. Learning results of the mean and worst losses on human-operated single-agent environment and eq. (18) with $H = 1$: since the worst loss was not needed to be minimized in this configuration, SPMO+MO ignored it to prioritize the mean loss.

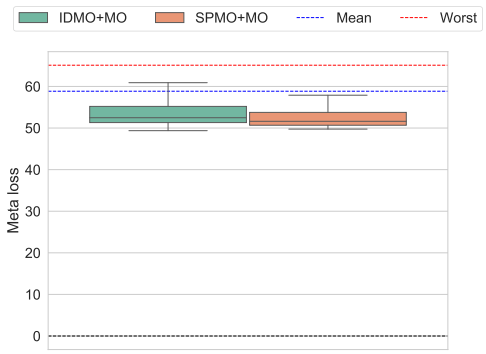


FIGURE 17. The comparison of the scores on human-operated single-agent environment and eq. (18) with $H = 1$: the scores of both SPMO+MO and IDMO+MO were reduced more in the cases of the maen loss and the worst loss; namely, the proposed methods leads to a better Pareto solution than the conventional method.

proposed methods reduce the meta-objective more than Mean and Worst. This indicates that the proposed methods can obtain the learning results with less meta loss by using the intermediate Pareto solution between Mean and Worst. Since Mean lowers the meta loss more than Worst, α converged to a value that emphasizes mean loss, is reasonable.

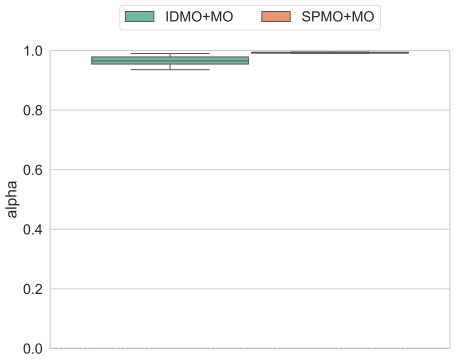


FIGURE 18. Learning results of α on multi-agent environment and eq. (18) with $H = 10$: α was trained to be close to 1; the worst loss was emphasized in the long-term prediction in this environment, as in the case of the human-operated single-agent environment.

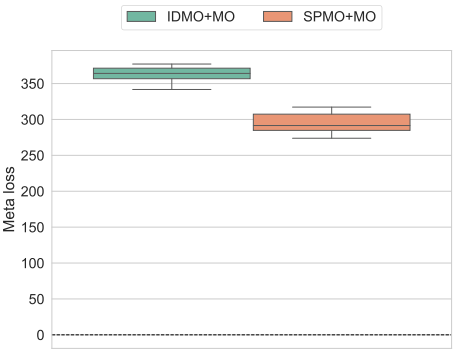
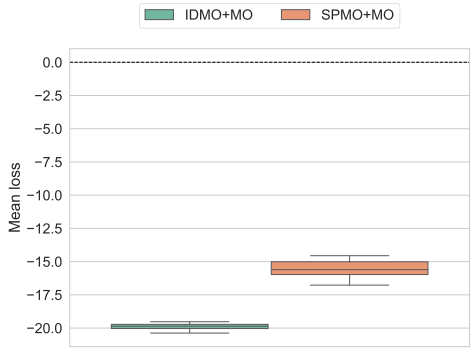


FIGURE 19. The results of the scores on multi-agent environment and eq. (18) with $H = 10$: the score was reduced more in the case of SPMO+MO than in the case of IDMO+MO, as in the case of the human-operated single-agent environment.

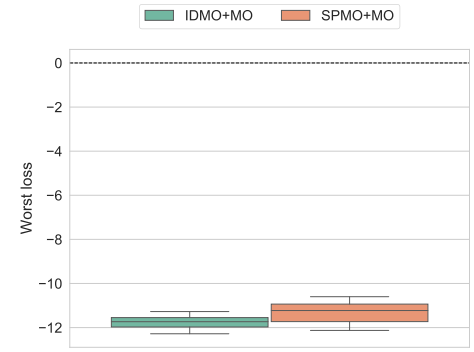
2) MULTI-AGENT ENVIRONMENT

The learning results in the multi-agent environment are shown below. As well as the case of the human-operated environment, the results for the time-horizon $H = 10$ depicted in Fig. 18 obtained $\alpha \simeq 1$. In addition, as shown in Fig.19, SPMO+MO could minimize the meta-objective much more than IDMO+MO. The reason for this result is also the same as for the previous environment: i.e. SPMO+MO acquired the generalized model by appropriate suppression of over-learning confirmed from a small gap between the mean and worse losses in Fig.20, while IDMO+MO did not. Note that both losses were smaller in IDMO+MO, but they were computed for the training data. In addition, Fig. 21 shows that, only SPMO+MO reduces the meta-objective to the same level as Worst.

On the other hand, even under the meta-objective with $H = 1$, α was adjusted toward 1 to emphasize the worst loss in both methods (see Fig. 22). As a result, Figs. 23 and 24 indicate that SPMO+MO and IDMO+MO obtained comparable performance. Fig. 25 shows that, both proposed methods reduce the meta-objective slightly more than Mean



(a) Mean loss



(b) Worst loss

FIGURE 20. Learning results of the mean and worst losses on multi-agent environment and eq. (18) with $H = 10$: SPMO+MO succeeded in keeping the difference between the mean and worst losses smaller than that of IDMO+MO, as in the case of the human-operated single-agent environment.

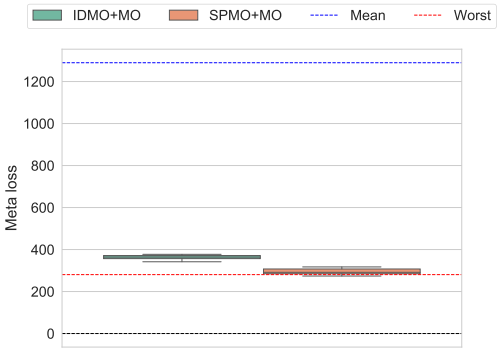


FIGURE 21. The comparison of the scores on multi-agent environment and eq. (18) with $H = 10$: the score of SPMO+MO was almost the same level as that of in the cases of the worst loss; namely, the results suggest that convergence to the worst loss was the optimal solution.

and Worst. Contrary to the case of the human-operated single-agent environment, since Worst lowers the meta loss more than Mean, α converged to a value that emphasizes worst loss, is reasonable.

The reason why α was optimized to be close to 1 even with $H = 1$ comes from the fact that this environment is partially observable (i.e. POMDP). Specifically, state transitions that occur in response to unobservable states are unavoidably expressed as uncertainty, hence the uncertainty of state

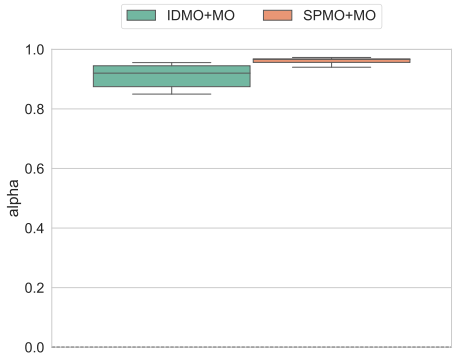


FIGURE 22. Learning results of α on multi-agent environment and eq. (18) with $H = 1$: α was optimized towards 1 unlike the case of the human-operated single-agent environment; this result suggested that the multi-agent environment was with high uncertainty and required the larger variance to cover it.

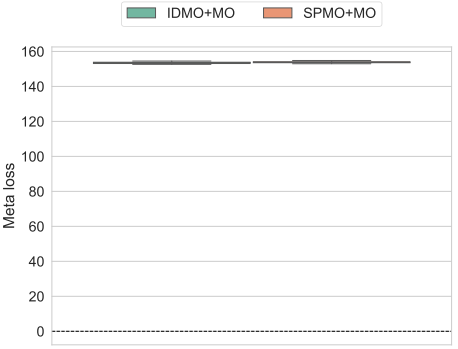


FIGURE 23. The results of the scores on multi-agent environment and eq. (18) with $H = 1$: the scores obtained from the meta-optimization were comparable for both methods.

transitions is inherently large in POMDP. The model trained with the mean loss cannot capture this uncertainty, and therefore it lacks generality to the validation and test data even if it is consistent with the meta-objective. To reduce the number of unexpected events as much as possible, the worst loss can be useful to make the variance of the model wider.

D. DISCUSSION

As investigated above, the optimal hyperparameters that lead to the preferred solution depend not only on the meta-objective but also on the contents of the dataset and the model architecture, hence it is not infeasible to give them analytically in advance. The proposed meta-optimization method based on the policy gradient allows us to obtain the preferred solution by adjusting the hyperparameters in a data-driven manner at the same time as learning the model. In addition, adjusting all hyperparameters will have little effect, namely we must make sure that which hyperparameters have the capability to find the Pareto frontier, as like α in SPMO.

One of the concerns is the exploration performance of the meta-optimization methods. Although the augmented weighted Tchebycheff scalarization theoretically guarantees

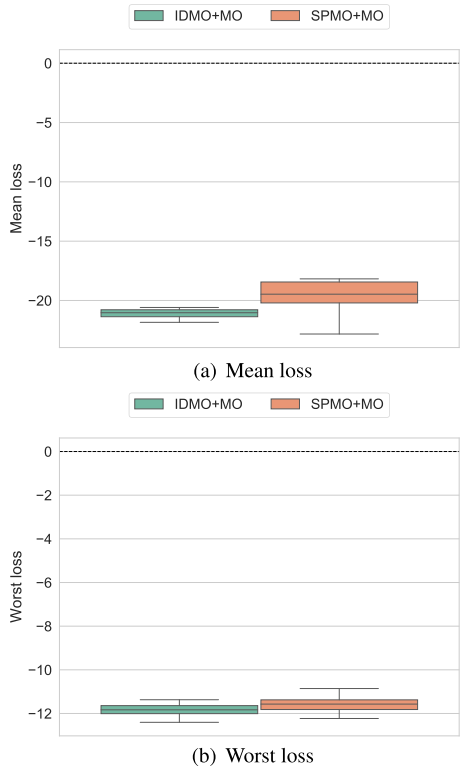


FIGURE 24. Learning results of the mean and worst losses on multi-agent environment and eq. (18) with $H = 1$: (a) the mean loss was kept at the same level for both methods; (b) the worst loss was also comparable for both methods.

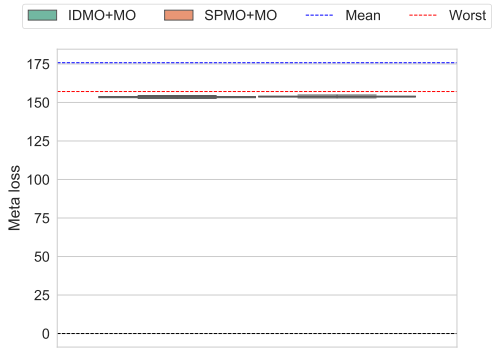


FIGURE 25. The comparison of the scores on multi-agent environment and eq. (18) with $H = 1$: the scores of both SPMO+MO and IDMO+MO were reduced slightly in the cases of the maen loss and the worst loss; namely, the proposed methods leads to a little better Pareto solution than the conventional method.

the reachability of all Pareto solutions, optimizing the hyperparameters with the policy gradient method may lead to local solutions in the meta-objective. To address this open issue, adding an auxiliary term to the meta-objective function defined in eq. (12) to facilitate the exploration (e.g. entropy of the meta-policy [49]) may increase the reachability of the global optimal solution.

Another concern is the effect of the variation of the optimization target on the stochastic model learning. In the proposed method, the loss function, which is defined as a MOO problem used in stochastic model learning, is modified at each epoch due to the simultaneous low-level learning and meta-optimization. As in curriculum learning [50], adaptive changes in the optimization target sometimes provide opportunity to escape from the local solutions, but vice versa. In combination with the exploration facilitation described above, this problem may be solved in practice, but deeper investigation is necessary.

Finally, this paper has developed the meta-optimization method starting from model learning for the model-based RL. Although this model learning is done in an offline manner with datasets already constructed, the model-based RL often involves planning and adding data using the model even in the process of learning [51], [52]. How the proposed method affects such online applications remains an open issue.

VI. CONCLUSION

This paper proposed a stochastic model learning method that is adjustable the bias-variance trade-off of the stochastic model according to higher-level objective. The proposed method consists of the loss function derived from the two-step MOO problem with inter-data and statistic-perspective objectives, and the meta-optimization of the hyperparameter in the loss function. Specifically, we first pointed out that the conventional loss for model learning is described as the inter-data MOO problem. The inter-data MOO was reformulated as the multiple single objective optimizations using the augmented weighted Tchebycheff scalarization. Furthermore, by applying the augmented weighted Tchebycheff scalarization again to the weighted sum of the mean and worst losses naturally obtained above, we defined the loss function as the stochastic-perspective MOO problem. The meta-optimization method was newly developed to balance the bias and the variance of the resulting stochastic model by adjusting the hyperparameter in the proposed loss function. Inspired by the policy-gradient method, that can be accomplished simultaneously with model learning only during a single trial with two model learners. The proposed method was applied to the human-operated single-agent and multi-agent environments with different types of uncertainty. First, the weighted sum of the mean loss and the worst loss was used as the meta-objective. The results showed that the hyperparameter was able to be adjusted according to the weight between the mean and worst loss with positive correlation. Next, the long-term prediction accuracy was used as another practical meta-objective. We found that the proposed method can improve the long-term prediction accuracy by revealing the best balance of the bias-variance trade-off and avoiding overfitting to the training data.

As mentioned in the discussion, the exploration performance should be guaranteed in order to acquire global solution. The analysis of the learning dynamics during meta-optimization is not completed yet. Furthermore, by extending the proposed method in an online learning manner, it can be integrated with model-based RL in the near future.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[2] G. Bingjing, H. Jianhai, L. Xiangpan, and Y. Lin, "Human–robot interactive control based on reinforcement learning for gait rehabilitation training robot," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419839584.

[3] H. Oliff, Y. Liu, M. Kumar, M. Williams, and M. Ryan, "Reinforcement learning for facilitating human-robot-interaction in manufacturing," *J. Manuf. Syst.*, vol. 56, pp. 326–340, Jul. 2020.

[4] K. Zhang, T. Sun, Y. Tao, S. Genc, S. Mallya, and T. Basar, "Robust multi-agent reinforcement learning with model uncertainty," in *Proc. NeurIPS*, 2020, pp. 1–20.

[5] T. Aotani, T. Kobayashi, and K. Sugimoto, "Bottom-up multi-agent reinforcement learning by reward shaping for cooperative-competitive tasks," *Int. J. Speech Technol.*, vol. 51, no. 7, pp. 4434–4452, Jul. 2021.

[6] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–10.

[7] W. Zhang, O. Bastani, and V. Kumar, "MAMPS: Safe multi-agent reinforcement learning via model predictive shielding," 2019, *arXiv:1910.12639*.

[8] D. D. Fan, A.-A. Agha-Mohammadi, and E. A. Theodorou, "Deep learning tubes for tube MPC," 2020, *arXiv:2002.01587*.

[9] B. T. Lopez, J.-J.-E. Slotine, and J. P. How, "Dynamic tube MPC for nonlinear systems," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 1655–1662.

[10] S. Gros and M. Zanon, "Safe reinforcement learning with stability & safety guarantees using robust MPC," 2020, *arXiv:2012.07369*.

[11] R. Pasquier and I. F. C. Smith, "Robust system identification and model predictions in the presence of systematic uncertainty," *Adv. Eng. Informat.*, vol. 29, no. 4, pp. 1096–1109, Oct. 2015.

[12] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4754–4765.

[13] H. Xu, C. Caramanis, and S. Mannor, "Robust regression and lasso," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3561–3574, Jul. 2010.

[14] D. Bertsimas and O. Nohadani, "Robust maximum likelihood estimation," *Informs J. Comput.*, vol. 31, no. 3, pp. 445–458, 2019.

[15] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, no. 1, pp. 1–58, 1992.

[16] B. Neal, S. Mittal, A. Baratin, V. Tantia, M. Scicluna, S. Lacoste-Julien, and I. Mitliagkas, "A modern take on the bias-variance tradeoff in neural networks," 2018, *arXiv:1810.08591*.

[17] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 32, pp. 15849–15854, Aug. 2019.

[18] V.-J. Aguilera-Rueda, N. Cruz-Ramírez, and E. Mezura-Montes, "Data-driven Bayesian network learning: A bi-objective approach to address the bias-variance decomposition," *Math. Comput. Appl.*, vol. 25, no. 2, p. 37, Jun. 2020.

[19] S. Paul, V. Kurin, and S. Whiteson, "Fast efficient hyperparameter tuning for policy gradients," 2019, *arXiv:1902.06583*.

[20] D. Salinas, H. Shen, and V. Perrone, "A quantile-based approach for hyperparameter transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8438–8448.

[21] G. Chen, "Merging deterministic policy gradient estimations with varied bias-variance tradeoff for effective deep reinforcement learning," 2019, *arXiv:1911.10527*.

[22] R. E. Steuer and E.-U. Choo, "An interactive weighted Tchebycheff procedure for multiple objective programming," *Math. Program.*, vol. 26, no. 3, pp. 326–344, 1983.

[23] K. Dächert, J. Gorski, and K. Klamroth, "An adaptive augmented weighted Tchebycheff method to solve discrete, integer-valued bicriteria optimization problems," Dept. Math., Univ. Wuppertal, Wuppertal, Germany, Tech. Rep. BUWAMNA-OPAP 10/06, 2010.

[24] J. Vanschoren, "Meta-learning: A survey," 2018, *arXiv:1810.03548*.

[25] Y. Li, Y. Yang, W. Zhou, and T. Hospedales, "Feature-critic networks for heterogeneous domain generalization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3915–3924.

[26] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "MetaReg: Towards domain generalization using meta-regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 998–1008.

[27] J. Grabocka, R. Scholz, and L. Schmidt-Thieme, "Learning surrogate losses," 2019, *arXiv:1905.10108*.

[28] S. Bechtle, A. Molchanov, Y. Chebotar, E. Grefenstette, L. Righetti, G. Sukhatme, and F. Meier, "Meta learning via learned loss," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4161–4168.

[29] C. Huang, S. Zhai, W. Talbott, M. B. Martin, S.-Y. Sun, C. Guestrin, and J. Susskind, "Addressing the loss-metric mismatch with adaptive loss alignment," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2891–2900.

[30] W. Zhou, Y. Li, Y. Yang, H. Wang, and T. M. Hospedales, "Online meta-critic learning for off-policy actor-critic methods," 2020, *arXiv:2003.05334*.

[31] V. Veeriah, M. Hessel, Z. Xu, J. Rajendran, R. L. Lewis, J. Oh, H. van Hasselt, D. Silver, and S. Singh, "Discovery of useful questions as auxiliary tasks," in *Proc. NeurIPS*, 2019, pp. 9310–9321.

[32] S. Gonzalez and R. Miikkulainen, "Improved training speed, accuracy, and data utilization through loss function optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.

[33] R. Houthooft, R. Y. Chen, P. Isola, B. C. Stadie, F. Wolski, J. Ho, and P. Abbeel, "Evolved policy gradients," 2018, *arXiv:1802.04821*.

[34] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castañeda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J. Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel, "Human-level performance in 3D multiplayer games with population-based reinforcement learning," *Science*, vol. 364, no. 6443, pp. 859–865, May 2019.

[35] M. Feurer, J. Springenberg, and F. Hutter, "Initializing Bayesian hyperparameter optimization via meta-learning," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 1–8.

[36] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019.

[37] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.

[38] P. Domingos, "A unified bias-variance decomposition," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 231–238.

[39] T. Heskes, "Bias/variance decompositions for likelihood-based estimators," *Neural Comput.*, vol. 10, no. 6, pp. 1425–1433, Aug. 1998.

[40] H. J. Bierens, "Information criteria and model selection," Penn State Univ., State College, PA, USA, Tech. Rep., 2004. [Online]. Available: https://faculty.wcas.northwestern.edu/~lchrist/workshop/Portugal/VAR/assignment2/INFORMATIONCRIT.pdf

[41] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2, no. 3. Cambridge, MA, USA: MIT Press, 2006.

[42] N. Hansen and S. Kern, "Evaluating the CMA evolution strategy on multimodal test functions," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 2004, pp. 282–291.

[43] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002.

[44] A. Engau, "Proper efficiency and tradeoffs in multiple criteria and stochastic optimization," *Math. Oper. Res.*, vol. 42, no. 1, pp. 119–134, Jan. 2017.

[45] T. K. Ralphs, M. J. Saltzman, and M. M. Wiecek, "An improved algorithm for solving biobjective integer programs," *Ann. Oper. Res.*, vol. 147, no. 1, pp. 43–70, Oct. 2006.

[46] W. E. L. Ilboudo, T. Kobayashi, and K. Sugimoto, "Robust stochastic gradient descent with student-t distribution based first-order momentum," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 16, 2020, doi: 10.1109/TNNLS.2020.3041755.

[47] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," 2016, *arXiv:1606.01540*.

[48] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," 2017, *arXiv:1706.02275*.

[49] J. Achiam and S. Sastry, "Surprise-based intrinsic motivation for deep reinforcement learning," 2017, *arXiv:1703.01732*.

[50] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[51] A. Strehl and M. Littman, "Online linear regression and its application to model-based reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1417–1424.
[52] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, "Model-based reinforcement learning via meta-policy optimization," in *Proc. Conf. Robot Learn.*, 2018, pp. 617–629.

TAKUMI AOTANI (Graduate Student Member, IEEE) received the B.S. degree from the National Institute of Technology, Maizuru College, Kyoto, Japan, in 2017, and the M.S. degree from the Nara Institute of Science and Technology, Nara, Japan, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include multi-agent reinforcement learning and safe reinforcement learning for autonomous robots.

TAISUKE KOBAYASHI (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from Nagoya University, Aichi, Japan, in 2012, 2014, and 2016, respectively. From 2018 to 2019, he was a Visiting Scholar with the Technical University of Munich, Munich, Germany. Since 2016, he has been an Assistant Professor with the Nara Institute of Science and Technology, Nara, Japan. Since 2020, he has been a JST PRESTO Researcher. His research interests include the locomotion control by intelligent systems and autonomous robotics with reinforcement learning.

KENJI SUGIMOTO (Member, IEEE) received the M.S. and Ph.D. degrees from Kyoto University, in 1982 and 1989, respectively. After working with Mitsubishi Electric Corporation, he became an Assistant Professor with Kyoto University, in 1985. Then, he became an Associate Professor with Okayama University and Nagoya University. Since 1999, he has been a Professor with the Nara Institute of Science and Technology. His current research interests include control theory and system science. He is a member of SICE and ISCIE.