Towards Applicable Reinforcement Learning: Improving the Generalization and Sample Efficiency with Policy Ensemble

Zhengyu Yang[1*], **Kan Ren**[2], **Xufang Luo**[2], **Minghuan Liu**[1], **Weiqing Liu**[2], **Jiang Bian**[2], **Weinan Zhang**[1] and **Dongsheng Li**[2]
[1]Shanghai Jiao Tong University
[2]Microsoft Research
{yzydestiny, minghuanliu, wnzhang}@sjtu.edu.cn,
{kan.ren, xufluo, weiqing.liu, jiang.bian, dongsli}@microsoft.com

Abstract

It is challenging for reinforcement learning (RL) algorithms to succeed in real-world applications like financial trading and logistic system due to the noisy observation and environment shifting between training and evaluation. Thus, it requires both high sample efficiency and generalization for resolving real-world tasks. However, directly applying typical RL algorithms can lead to poor performance in such scenarios. Considering the great performance of ensemble methods on both accuracy and generalization in supervised learning (SL), we design a robust and applicable method named Ensemble Proximal Policy Optimization (EPPO), which learns ensemble policies in an end-to-end manner. Notably, EPPO combines each policy and the policy ensemble organically and optimizes both simultaneously. In addition, EPPO adopts a diversity enhancement regularization over the policy space which helps to generalize to unseen states and promotes exploration. We theoretically prove EPPO increases exploration efficacy, and through comprehensive experimental evaluations on various tasks, we demonstrate that EPPO achieves higher efficiency and is robust for real-world applications compared with vanilla policy optimization algorithms and other ensemble methods. Code and supplemental materials are available at https://seqml.github.io/eppo.

1 Introduction

Compared with simple simulation tasks, it is more difficult for RL algorithms to succeed in real-world applications. First, the observation contains much noise and the sampling cost is more expensive in real-world applications. Second, the environment shifts between the training and evaluation due to the complexity of the real world. For instance, in financial trading, the noise in the imperfect market information puts forward high requirements on the sample efficiency of the algorithm, and the volatile market requires the algorithms not to overfit to the training environment and retain the ability to generalize to unseen states during evaluation. However, typical RL algorithms cannot achieve satisfactory performance in these applications. Motivated by the superior performance of ensemble methods on improving the accuracy and generalization ability in SL especially for small datasets, we resort to ensemble methods to fulfill the aforementioned requirements. In our work, we focus on policy ensemble, which is an integration of a set of *sub-policies*, instead of value function ensemble for some reasons listed below. i) Value-based methods perform worse than policy-based methods in noisy applications like MOBA games [Ye *et al.*, 2020], card games [Yang *et al.*, 2022; Li *et al.*, 2020] and financial trading [Fang *et al.*, 2021]. ii) Previous ensemble techniques for RL are mainly applied to SL components, like environment dynamics modeling [Kurutach *et al.*, 2018] and value function approximation [Anschel *et al.*, 2017]. iii) Policy learning in RL algorithms is critical which is more different from SL. But it is not well studied and thus is worth exploring.

Notice that in many real-world applications such as those mentioned above, Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017] is always the first choice of the underlying RL algorithm due to its excellent and stable performances. Therefore, in order to derive an applicable RL algorithm, in this paper, we take PPO as our backbone, and propose a simple yet effective policy ensemble method named Ensemble Proximal Policy Optimization (EPPO). EPPO rigorously treats ensemble policy learning as a first class problem to explicitly address: i) what is the reasonable yet effective policy ensemble strategy in deep RL and ii) how it helps to improve the performance of policy learning.

Some existing works for policy ensemble aim to attain a diverse set of policies through *individually* training various policies and simply aggregating them ex post factor [Wiering and Van Hasselt, 2008; Duell and Udluft, 2013; Saphal *et al.*, 2020], which has few guarantees to improve the overall performance due to the neglection of the cooperation among different sub-policies. The other works incorporate the divide-and-conquer principle to divide the state space, and derive a set of diverse sub-policies accordingly [Ghosh *et al.*, 2017; Goyal *et al.*, 2019; Ren *et al.*, 2021]. But the difficulty in di-

*The work was conducted during Zhengyu Yang's internship at Microsoft Research. The corresponding author is Kan Ren.

viding the state space and the unawareness of the sub-policy on the whole state space may significantly hurt the performance especially in *deep* RL. Furthermore, whether and how the ensemble method would benefit policy optimization still remain unsolved and require additional attention.

In contrast, EPPO resolves the ensemble policy learning from two aspects. On one side, we argue that ensemble learning and policy learning should be considered as a whole organic system to promote the cooperation among the sub-policies and guarantee the ensemble performance. Thus, EPPO combines sub-policy training and decision aggregation as a whole and optimizes them under a unified ensemble-aware loss. To fully exploit the data and improve sample efficiency, sub-policies are also optimized with the data collected by the ensemble policy which aggregates all the co-training sub-policies for final decision. Furthermore, we theoretically prove that the decision aggregation of co-training sub-policies helps in efficient exploration thus improves the sample efficiency. On the other side, considering that ensemble methods benefit from the diversity among sub-policies and it is difficult to divide the state space to train diverse policies reasonably, EPPO incorporates a diversity enhancement regularization within the policy space to guarantee the diversity and further improve the ensemble performance. We empirically found that it can improve policy generalization in real-world applications because the diversity enhancement regularization prevents the sub-policies from collapsing into a singular mode or over-fitting to the training environment, which retains the ability of the ensemble policy to generalize to unseen states.

In a nutshell, the main contributions of the work are threefold:

- We propose a simple yet effective ensemble strategy in ensemble policy learning and prove that aggregating the co-training sub-policies can promote policy exploration and improve sample efficiency.
- To the best of our knowledge, EPPO is the first work that adopts the diversity enhancement regularization to attain a diverse set of policies for policy ensemble.
- Demonstrated by the experiments on grid-world environments, Atari benchmarks and a real-world application, EPPO yields better sample efficiency and the diversity enhancement regularization also provides a promising improvement on policy generalization.

2 Background

2.1 Preliminaries

Sequential decision making process can be formulated as a Markov decision process (MDP), represented by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, p_0, r \rangle$. $\mathcal{S} = \{s\}$ is the space of the environment states. $\mathcal{A} = \{a\}$ is the action space of the agent. $p(s_{t+1}|s_t, a_t) : \mathcal{S} \times \mathcal{A} \mapsto \Omega(\mathcal{S})$ is the dynamics model, where $\Omega(\mathcal{S})$ is the set of distributions over $\mathcal{S}$. The initial state s_0 of the environment follows the distribution $p_0 : S \mapsto \mathbb{R}$. $r(s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function. The objective is learning a policy π to maximize the cumulative reward $\eta(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} r(s_t, a_t)\right]$ where τ is the trajectory sampled by π. In this paper, we consider discrete control tasks where $\mathcal{A}$ is limited and discrete.

2.2 Related Works

Ensemble in RL The recent works applying ensemble methods in RL are mainly focusing on environment dynamics modeling and value function approximation. For environment dynamics modeling, several environment models are used to reduce the model variance [Chua *et al.*, 2018] and stabilize the model-based policy learning [Kurutach *et al.*, 2018]. As for value function approximation, Q-function ensemble is popular in alleviating the over-estimation [Anschel *et al.*, 2017], encouraging exploration [Lee *et al.*, 2021] and realizing conservative policy learning in offline reinforcement learning [Wu *et al.*, 2021].

Nevertheless, the mechanism behind environment dynamics modeling and value function approximation is similar to SL and it has a huge gap with policy learning in RL. Among the existing works on ensemble policy learning, some works follow the technique used in SL and simply aggregate individually trained policies ex post facto. To generate a set of diverse sub-policies, different weight initialization [Faußer and Schwenker, 2015; Duell and Udluft, 2013], training epochs [Saphal *et al.*, 2020], or RL algorithms [Wiering and Van Hasselt, 2008] are used. Compared with SL tasks, RL agents must take a sequence of decisions instead of making a one-step prediction, which makes the cooperation among sub-policies more important to attain a good ensemble. Without considering policy ensemble and policy learning as a whole optimization problem, the cooperation among sub-policies can be neglected, so that these methods have few guarantees to improve the overall performance. The other works incorporate the divide-and-conquer principle to divide the state space and derive a set of specialized policies accordingly, and then these policies are aggregated to solve the original task, which coincides with the idea of mixture-of-experts (MOE) [Jacobs *et al.*, 1991]. The essential of the MOE is how to deliver data and obtain a set of specialized policies (i.e. experts) which focus on different regions of the state space. EPPO, which proposes a new method (i.e. diversity enhancement regularization) to derive an ensemble of experts and shows better performance in experiments, is also a special case under the paradigm of MOE. To divide the state space, DnC [Ghosh *et al.*, 2017] heuristically divides the whole task into several sub-tasks based on the clustering of the initial states while ComEns [Goyal *et al.*, 2019] and PMOE [Ren *et al.*, 2021] learn a division principle based on information theory and Gaussian mixture model respectively. However, it is hard to divide the state space in many environments and the unreasonable division can damage the performance. Moreover, the unawareness of the whole state space of the sub-policies caused by the division may significantly hurt the performance especially in *deep* RL scenarios and result in poor ensemble effectiveness.

Diversity Enhancement Diversity enhancement, which aims at deriving a set of diverse policies, is mainly used in population-based RL. To enhance the diversity, KL divergence [Hong *et al.*, 2018], maximum mean discrepancy [Masood and Doshi-Velez, 2019] and determinantal point pro-

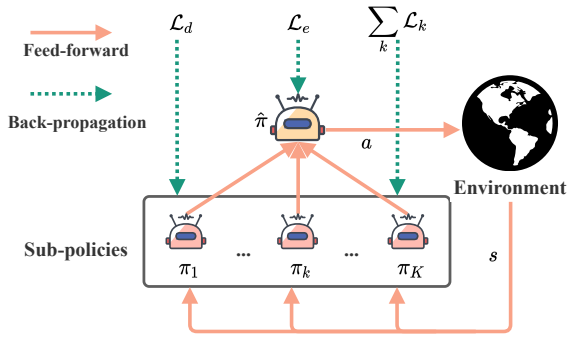

Figure 1: Framework of EPPO. Only the ensemble policy $\hat{\pi}$ is used to interact with the environment while all the sub-policies are updated simultaneously based on the same collected data.

cess [Parker-Holder *et al.*, 2020] are adopted as bonus during training process. In policy ensemble, the diversity among sub-policies is also important, but almost all the existing methods apply state space division to enhance the diversity, which may hurt the performance as mentioned before. Thus, we impose the diversity enhancement regularization on the policy space to guarantee the diversity among the sub-policies.

3 Ensemble Proximal Policy Learning

In this section, we first motivate our design in EPPO on the training of sub-policies and data collection, and then introduce the details of the learning method. The overview of the architecture is illustrated in Figure 1.

3.1 Policy Ensemble

As widely used in the literature of ensemble methods [Zhou *et al.*, 2002; Anschel *et al.*, 2017], the approximated function is aggregated by a set of base components, each of which can be optimized and work individually in the target task. Thereafter, we consider maintaining K sub-policies $\{\pi_{\theta_1}, \pi_{\theta_2}, \ldots, \pi_{\theta_K}\}$. For brevity, we denote π_k to represent the sub-policy parameterized by θ_k. Then, the ensemble policy $\hat{\pi}$ can be derived through mean aggregation over the sub-policies. Formally, for a given state s, the ensemble policy $\hat{\pi}(\cdot|s)$ is calculated as the arithmetic mean of the sub-policies:

$$\hat{\pi}(\cdot|s) = \frac{1}{K}\sum_{k=1}^{K}\pi_k(\cdot|s)\ . \tag{1}$$

Note that the parameters of the ensemble policy $\hat{\pi}$ are exactly the whole set of parameters of sub-policies $\{\theta_k\}_{k=1}^K$. In RL tasks, sample efficiency is a key problem and we would expect that the performance improvement of the designed ensemble method comes from the algorithm itself instead of the more trajectories sampled by more sub-policies. Thus, only the ensemble policy $\hat{\pi}$ is allowed for data collection in EPPO and the agent samples the action $a \sim \hat{\pi}(\cdot|s)$ from the ensemble policy $\hat{\pi}$ when interacting with the environment. Then the trajectories sampled by the ensemble policy $\hat{\pi}$ will be further used for updating the sub-policies.

3.2 Policy Optimization

The previous ensemble works in SL also motivate a fact that better sub-models lead to better empirical results [Zhang *et al.*, 2020], thus we apply PPO to maximize the expected return (i.e., $\eta(\pi_k), 1 \le k \le K$) of the sub-policies and the loss is defined as:

$$\mathcal{L}_k(\pi_k) = \mathbb{E}_{\hat{\pi}'}\left[\sum_{t=0}^{T}\mu\mathrm{KL}\left[\hat{\pi}'(\cdot|s_t), \pi_k(\cdot|s_t)\right] - \frac{\pi_k(a_t|s_t)}{\hat{\pi}'(a_t|s_t)}\hat{A}_{\hat{\pi}'}(t)\right], \tag{2}$$

where $\hat{\pi}'$ is the ensemble policy, μ is an adaptive penalty parameter to constrain the size of the policy update, and $\hat{A}_{\hat{\pi}'}(t) = \hat{A}_{\hat{\pi}'}(s_t, a_t)$ is the advantage function estimated by a generalized advantage estimator (GAE) [Schulman *et al.*, 2015] which describes how much better the action is than others on average. It is worth noting the data used for optimization is collected by the ensemble policy $\hat{\pi}$, and we only optimize the parameters of sub-policies through policy gradient, which does not impose any additional sample cost in the environment as that of a single policy.

Though simply aggregating the predictions of the sub-models in the ensemble has shown effectiveness in improving the performance for supervised tasks [Zhou *et al.*, 2018], there is less evidence showing that the aggregation of the sub-policies can improve decision making, because of the large gap between the essential of RL and SL, such as: i) there may be more than one optimal action at the current state in RL while there is only one ground truth for SL; ii) the aggregation of some well-behaved sub-policies may derive undesirable action distribution and lead to bad states, especially when there are many different ways to handle the task and the sub-policies are separately optimized. Thus, it is necessary to take the cooperation among sub-policies into consideration and optimize them in a consistent learning paradigm. In EPPO, we incorporate an ensemble-aware loss to encourage the cooperation among the sub-policies and ensure a well-behaved ensemble policy $\hat{\pi}$. The definition of ensemble-aware loss which optimizes the ensemble policy $\hat{\pi}$ by PPO is

$$\mathcal{L}_e(\hat{\pi}) = \mathbb{E}_{\hat{\pi}'}\left[\sum_{t=0}^{T}\mu\mathrm{KL}\left[\hat{\pi}'(\cdot|s_t), \hat{\pi}(\cdot|s_t)\right] - \frac{\hat{\pi}(a_t|s_t)}{\hat{\pi}'(a_t|s_t)}\hat{A}_{\hat{\pi}'}(t)\right]. \tag{3}$$

Taking Eq. (1) into Eq. (3), we can update the ensemble policy by updating all sub-policies in a unified behavior under the same target. To some extent, the ensemble-aware loss serves as a regularization that may promote the ensemble performance at the cost of the performance of sub-policies.

However, there is a potential risk of mode collapse in policy ensemble that all the sub-policies converge to a single policy, which makes policy ensemble useless. In our method, the problem is even worse because all the sub-policies share a similar training paradigm, which makes these sub-policies tend to behave similarly. Moreover, EPPO randomly chooses a sub-policy for action sampling at each step (Eq.(1)), so the diversity among sub-policies should promote the exploration of the ensemble policy. To this end, we propose a diversity enhancement regularization to prevent all sub-policies from collapsing into a singular mode and ensure diverse sub-policies to further improve the ensemble performance. Intuitively, in

order to enhance the diversity, the regularization should encourage the action distributions proposed by different sub-policies to be orthogonal with each other. Specifically, for discrete action space, the diversity enhancement regularization adopted in EPPO is defined as

$$\mathcal{L}_d = \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} \sum_{a} \pi_i(a|s)\pi_j(a|s). \quad (4)$$

We note that there are many optional metrics to encourage the diversity among sub-policies such as KL divergence [Hong *et al.*, 2018] and MMD [Masood and Doshi-Velez, 2019] in RL literature, yet we adopt Eq.(4) in EPPO due to its computationally efficiency [Li *et al.*, 2012].

In conclusion, the overall loss to minimize is defined as

$$\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}_k(\pi_k) + \alpha \mathcal{L}_e(\hat{\pi}) + \beta \mathcal{L}_d, \quad (5)$$

where α and β are the hyper-parameters.

3.3 Theoretical Analysis

Theorem 1 (Mean aggregation encourages exploration)**.** *Suppose π and $\{\pi_i\}_{1 \le i \le K}$ are sampled from $P(\pi)$, then the entropy of the ensemble policy $\hat{\pi}$ is no less than the entropy of the single policy in expectation, i.e., $\mathbb{E}_{\pi_1,\pi_2,\ldots,\pi_K}[\mathcal{H}(\hat{\pi})] \ge \mathbb{E}_{\pi}[\mathcal{H}(\pi)]$.*

The proof can be found in Appendix A in the supplemental materials. Theorem 1 illustrates that the ensemble policy $\hat{\pi}$ enjoys more effective exploration than the single policy in the policy learning procedure. Thus, aggregating the sub-policies during training can improve the sample efficiency. We have also observed the corresponding phenomenon in the experiments, which reflects the effectiveness of the mean aggregation operation for policy ensemble in our method.

4 Experiments

In our paper, we only consider discrete control tasks as it is commonly adopted in real-world scenario applications and the continuous control tasks can be discretized for the ease of optimization. To evaluate the performance of EPPO, we conduct experiments on Minigrid [Chevalier-Boisvert *et al.*, 2018], Atari games [Bellemare *et al.*, 2013] and financial trading [Fang *et al.*, 2021], which span the simulated tasks and real-world applications.

The experiments and the analysis in this section are led by the following two research questions (RQs). **RQ1**: Does our method achieve higher sample efficiency through policy ensemble? **RQ2**: Is the generalization performance of our method better than the other compared methods?

4.1 Compared Methods

We compare EPPO with the following baselines including two variants of EPPO.

- **PPO** [Schulman *et al.*, 2017] is a state-of-the-art policy optimization method which has been widely used in real-world applications [Fang *et al.*, 2021; Ye *et al.*, 2020].
- **PE** (Policy Ensemble) is based on a traditional policy ensemble method [Duell and Udluft, 2013] which trains K policies *individually* by PPO and then aggregates them. In our paper, we consider two aggregation operation, i.e., majority voting and mean aggregation, which are defined as $\hat{\pi}(a|s) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}((\arg\max_{a'} \pi_k(a'|s)) == a)$ and Eq. (1), respectively. We denote these two methods as **PEMV** and **PEMA** for short.
- **DnC** [Ghosh *et al.*, 2017] partitions the initial state space into K slices, and optimizes a set of policies each on different slices. During training, these policies are periodically distilled into a center policy that is used for evaluation.
- **ComEns** [Goyal *et al.*, 2019] uses an information-theoretic mechanism to decompose the policy into an ensemble of primitives and each primitive can decide for themselves whether they should act in current state.
- **PMOE** [Ren *et al.*, 2021] applies a routing function to aggregate the sub-policies and deliver the data to different sub-policies during optimization.
- **SEERL** [Saphal *et al.*, 2020] uses a learning rate schedule to get multiple policies in a round and selects a set of policies for ensemble according to performance and diversity.
- **EPPO** is our proposed method described above, which has two other variants for ablation study: **EPPO-Div** is the method *without* diversity enhancement regularization defined in Eq. (4) and **EPPO-Ens** is the method *without* ensemble-aware loss defined in Eq. (3).

Since we focus on policy ensemble, we omit Q-function ensemble methods like Sunrise [Lee *et al.*, 2021]. In all compared baselines, we take PPO as the base policy optimization method for fairness; besides, they have roughly the same number of parameters (i.e., K times the parameter size of PPO and we set $K = 4$ as default for all experiments) and the same number of samples collected in one training epoch.

4.2 Improved Efficiency on Minigrid

We first investigate whether EPPO can improve the sample efficiency. In this part, we consider two partial-observable environments with sparse reward in Minigrid [Chevalier-Boisvert *et al.*, 2018]: *Distributional-Shift* and *Multi-Room*, as shown in Figure 2, where the agent aims at reaching the given target position and a nonzero reward is provided only if the target position is reached. Specifically, the place of the second line of the lava in *Distributional-Shift* is reset, and the shape of *Multi-Room* is regenerated during the reset procedure. Due to the more complex structure and longer distance between the start position and the goal, *Multi-Room* is more difficult.

As shown in Figure 2, EPPO enjoys the best sample efficiency in both environments (**RQ1**). We notice that PEMA and PEMV fail (i.e., return $= 0$) in both environments while PPO can get better performance. And we conclude the failure from two perspectives. First, considering the number of samples needed for PPO to get a positive reward is large, PEMA and PEMV may need K times samples for a positive reward because they individually train K PPO policies. Second, due to the asynchrony among the sub-policies of PE method, the useful knowledge can be overwhelmed thus neglected due to

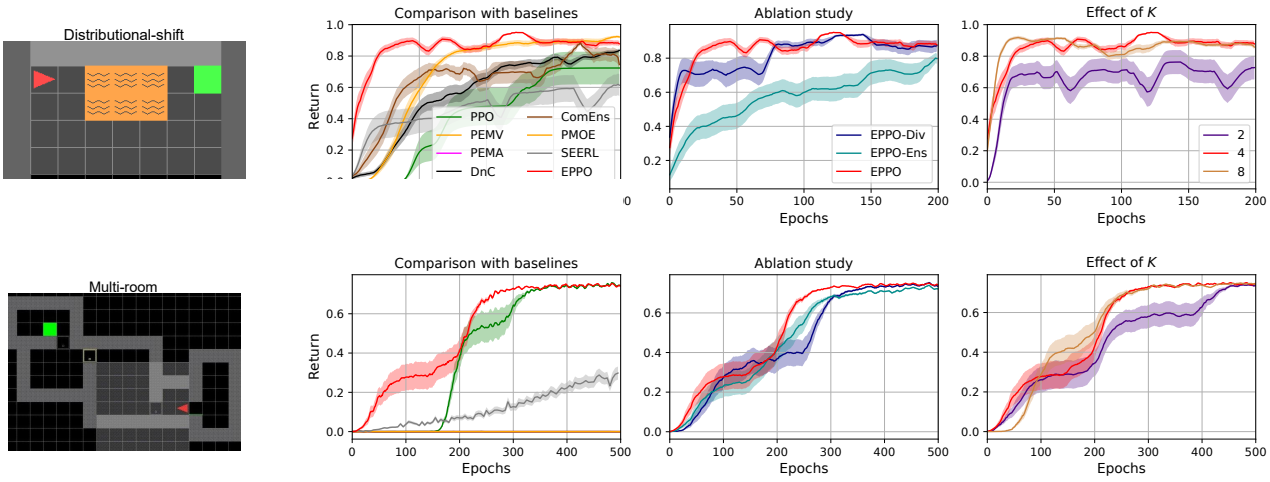

Figure 2: The first column gives a snapshot of the environments where the red triangle and green square represent the position and the goal of the agent respectively. Learning results on Minigrid conducted with 5 random seeds. The top and bottom rows show the information about *Distributional-Shift* and *Multi-Room*, respectively. **First column**: A snapshot of the environment where the red triangle and green square represent the position and the goal of the agent respectively. **Second column**: Learning curves of all the compared methods. **Third column**: Learning curves of EPPO and its variants. **Last column**: Learning curves of EPPO when K is set to different values.

the aggregation operation by the worthless knowledge. Their failure implies the necessity of an ensemble-aware loss and explains the rationality of sampling with ensemble policy $\hat{\pi}$. For SEERL, its final performance is even worse than that at the last epoch because the ex-post selection cannot ensure an improved performance, which further emphasizes the necessity of the joint optimization of the sub-policies. Moreover, the failure of DnC, ComEns and PMOE in *Multi-Room* environment can be attributed to the division operation on the state space that may not only unreasonably divide the space but also hinder the ability on exploration to the full environment. And comparing the performance of the methods based on state space division and EPPO, we find that EPPO consistently achieves better performance, which implies that a diversity enhancement regularization is a better choice for diversity enhancement in policy ensemble.

The ablation study in Figure 2 shows that EPPO outperforms its two variants, thus both of diversity enhancement regularization and ensemble-aware loss appear to be crucial for the superior performance of EPPO. In addition, the returns of both EPPO and its variants improve quickly at first, which confirms the result in Theorem 1 that mean aggregation encourages exploration. In the last column of Figure 2, we analyze the effect of using various number of sub-policies in EPPO. The results indicate that an extremely small K cannot lead to a good performance and the performance cannot be further improved by increasing K by a large margin. When K is a small value like 2, the sub-policies tend to have fewer overlaps in the action space, thus the mean aggregation operation is unable to extract the valuable information from sub-policies and leads to worse performance.

4.3 Comparative Evaluations on Atari Games

Having seen the superior performance of EPPO in environments with sparse reward, we also want to evaluate EPPO on more difficult and widely used benchmarks. Particularly, we follow the settings in [Saphal *et al.*, 2020] and choose four environments from Atari games as the testbed. As shown in Table 1, EPPO can still consistently achieve the best performance in 10M environment steps, suggesting a better sample efficiency (**RQ1**).

	Alien	Amidar	Pong	Seaquest
PPO	1174.6	283.8	20.8	1110.2
PEMV	678.0	74.3	6.9	364.2
PEMA	815.2	113.8	7.6	563.4
DnC	158.0	41.5	-21.0	185.0
ComEns	351.6	51.0	-20.7	504.2
PMOE	1488.2	247.3	3.0	1800.5
SEERL	1127.8	155.0	20.0	928.4
EPPO-Div	1173.2	304.6	19.4	1580.8
EPPO-Ens	1651.2	311.8	20.8	1816.6
EPPO	**1984.0**	**439.7**	**20.9**	**1881.2**

Table 1: Performance on Atari games at 10M interactions. All results represent the average over 100 episodes of 5 random training runs. Bold font represents the best results.

4.4 Generalizable Application: A Financial Trading Instance

To evaluate the generalization ability, we conduct experiments on order execution [Fang *et al.*, 2021] which is a fundamental yet challenging problem in financial trading. In order execution, the environments are built upon the historical transaction data, and the agent aims at fulfilling a trading order which specifies the date, stock id and the amount of stock needed to be bought or sold. In particular, the environments are usually formulated as training, validation and test phases each of which is corresponding to a specific time range. To be specific, *training environment* and *validation environment*

Phase	Dataset 1801-1908	
	# order	Time Period
Training	845,006	01/01/2018 - 31/12/2018
Validation	132,098	01/01/2019 - 28/02/2019
Test	455,332	01/03/2019 - 31/08/2019
Phase	Dataset 1807-2002	
	# order	Time Period
Training	854,936	01/07/2018 - 30/06/2019
Validation	163,140	01/07/2019 - 31/08/2019
Test	428,846	01/09/2019 - 29/02/2020

Table 2: The dataset statistics of financial order execution task.

are used for policy optimization and policy selection respectively. *Test environment* is unavailable during training. Due to the shift of macroeconomic regulation or other factors in different time, *test environment* may differ a lot to the *training* and *validation environment*. Thus, the selected policy has to make decisions in unfamiliar states during testing and the performance in *test environment* is a good surrogate evaluation of the generalization ability.

Following [Fang *et al.*, 2021], the reward is composed of price advantage (PA) and market impact penalty where PA encourages the policy to get better profit than a baseline strategy. Specifically, we take TWAP as the baseline strategy, which equally splits the order into T pieces and evenly executes the same amount of shares at each timestep during the whole time horizon. And an increase of 1.0 in PA can bring about a 0.5% annual return with 20% daily turnover rate. The averaged PA and reward of the orders in the *test environment* are taken as evaluation metrics in order execution. We conduct experiments on the two large datasets 1801-1908 and 1807-2002 published in [Fang *et al.*, 2021] and the statistics of the datasets can be found in Table 2.

The results of different methods are reported in Table 3. As expected, EPPO achieves the best performance in both PA and reward in two datasets, suggesting that our proposed method has great potential in generalizing to unseen states (**RQ2**). And we find that PEMV has a worse reward than PPO in 1801-1908, which implies individually training sub-policies has no guarantee on the ensemble performance, thus an ensemble-aware loss, i.e., L_e in Eq. (3) which can encourage the coordination among sub-policies, is necessary. In addition, the performance degradation of EPPO-Ens also illustrates the importance of the ensemble-aware loss. Moreover, from the comparison between EPPO and EPPO-Div, we find that the diversity enhancement regularization further improves the generalization performance. This phenomenon coincides with the observations in SL [Zhou *et al.*, 2002] that the diversity among sub-models can reduce the variance, alleviate the over-fitting problem and improve the generalization performance of the ensemble method.

To evaluate the effect of K, we conduct experiments when $K \in \{2, 4, 8\}$ in dataset 1801-1908 and the results are shown in Table 4. Similar to the experimental results in Minigrid, $K = 2$ leads to a poor performance, which can still be attributed to the difficulty of getting consensus during aggregation when K is small. In addition, a larger K does not always lead to better performance.

Dataset	1801-1908		1807-2002	
Metric	PA	Reward	PA	Reward
PPO	7.43	4.57	5.30	2.75
PEMV	7.47	4.41	6.03	3.44
PEMA	7.87	5.00	5.98	3.42
DnC	7.99	5.47	5.36	2.75
ComEns	7.70	4.79	4.59	1.32
PMOE	3.12	-0.03	3.09	1.43
SEERL	7.03	5.04	5.52	3.51
EPPO-Div	8.38	5.51	6.21	3.30
EPPO-Ens	6.38	3.87	5.51	3.51
EPPO	**8.82**	**5.99**	**6.31**	**3.57**

Table 3: Test performance on order execution task; the higher metric value means the better performance. The results are the average of all test orders over ten random training runs.

K	2	4	8
PA	7.49	**8.82**	8.64
Reward	4.42	**5.99**	5.83

Table 4: Results on different K.

4.5 Analysing the Policy Diversity

After showing the performance of diversity enhancement regularization in improving the sample efficiency and policy generalization, we further verify that the regularization does diversify the sub-policies. Motivated by [Hong *et al.*, 2018], we utilize the action disagreement (AD) to measure the diversity among sub-policies, which is defined as

$$\frac{\sum_{i,j} \sum_{s \in M} \mathbb{I}(\arg\max_a \pi_i(a|s) \neq \arg\max_a \pi_j(a|s))}{|M|K(K-1)}, \quad (6)$$

where M is a set of states. In dataset 1801-1908, the AD values of EPPO and EPPO-Div are 15.9% and 14.3%, respectively, which demonstrates the ability of diversity enhancement regularization in improving diversity.

5 Conclusion and Future Work

In this paper, we focus on ensemble policy learning and propose an end-to-end ensemble policy optimization framework called EPPO that combines sub-policy training and policy ensemble as a whole. In particular, EPPO updates all the sub-policies simultaneously under the ensemble-aware loss with a diversity enhancement regularization. We also provide a theoretical analysis of EPPO on improving the entropy of the policy which leads to better exploration. Extensive experiments on various tasks demonstrate that EPPO substantially outperforms the baselines for both sample efficiency and policy generalization performance. In the future, we plan to incorporate more flexible sub-policy ensemble mechanisms and dive deeper into the mechanism behind the ensemble policy learning.

References

[Anschel *et al.*, 2017] Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *ICML*, pages 176–185. PMLR, 2017.

[Bellemare *et al.*, 2013] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *JAIR*, 47:253–279, 2013.

[Chevalier-Boisvert *et al.*, 2018] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018. Accessed: 2021-08-30.

[Chua *et al.*, 2018] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *NeurIPS*, pages 4759–4770, 2018.

[Duell and Udluft, 2013] Siegmund Duell and Steffen Udluft. Ensembles for continuous actions in reinforcement learning. In *ESANN*. Citeseer, 2013.

[Fang *et al.*, 2021] Yuchen Fang, Kan Ren, Weiqing Liu, Dong Zhou, Weinan Zhang, Jiang Bian, Yong Yu, and Tie-Yan Liu. Universal trading for order execution with oracle policy distillation. *arXiv preprint arXiv:2103.10860*, 2021.

[Faußer and Schwenker, 2015] Stefan Faußer and Friedhelm Schwenker. Neural network ensembles in reinforcement learning. *Neural Processing Letters*, 41(1):55–69, 2015.

[Ghosh *et al.*, 2017] Dibya Ghosh, Avi Singh, Aravind Rajeswaran, Vikash Kumar, and Sergey Levine. Divide-and-conquer reinforcement learning. *arXiv preprint arXiv:1711.09874*, 2017.

[Goyal *et al.*, 2019] Anirudh Goyal, Shagun Sodhani, Jonathan Binas, Xue Bin Peng, Sergey Levine, and Yoshua Bengio. Reinforcement learning with competitive ensembles of information-constrained primitives. *arXiv preprint arXiv:1906.10667*, 2019.

[Hong *et al.*, 2018] Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. *arXiv preprint arXiv:1802.04564*, 2018.

[Jacobs *et al.*, 1991] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[Kurutach *et al.*, 2018] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. *arXiv preprint arXiv:1802.10592*, 2018.

[Lee *et al.*, 2021] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *ICML*, pages 6131–6141. PMLR, 2021.

[Li *et al.*, 2012] Nan Li, Yang Yu, and Zhi-Hua Zhou. Diversity regularized ensemble pruning. In *ECMLPKDD*, pages 330–345. Springer, 2012.

[Li *et al.*, 2020] Junjie Li, Sotetsu Koyamada, Qiwei Ye, Guoqing Liu, Chao Wang, Ruihan Yang, Li Zhao, Tao Qin, Tie-Yan Liu, and Hsiao-Wuen Hon. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv preprint arXiv:2003.13590*, 2020.

[Masood and Doshi-Velez, 2019] Muhammad A Masood and Finale Doshi-Velez. Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. *arXiv preprint arXiv:1906.00088*, 2019.

[Parker-Holder *et al.*, 2020] Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. *NeurIPS*, 33:18050–18062, 2020.

[Ren *et al.*, 2021] Jie Ren, Yewen Li, Zihan Ding, Wei Pan, and Hao Dong. Probabilistic mixture-of-experts for efficient deep reinforcement learning. *arXiv preprint arXiv:2104.09122*, 2021.

[Saphal *et al.*, 2020] Rohan Saphal, Balaraman Ravindran, Dheevatsa Mudigere, Sasikanth Avancha, and Bharat Kaul. Seerl: Sample efficient ensemble reinforcement learning. *arXiv preprint arXiv:2001.05209*, 2020.

[Schulman *et al.*, 2015] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

[Wiering and Van Hasselt, 2008] Marco A Wiering and Hado Van Hasselt. Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):930–936, 2008.

[Wu *et al.*, 2021] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*, 2021.

[Yang *et al.*, 2022] Guan Yang, Minghuan Liu, Weijun Hong, Weinan Zhang, Fei Fang, Guangjun Zeng, and Yue Lin. Perfectdou: Dominating doudizhu with perfect information distillation. *arXiv preprint arXiv:2203.16406*, 2022.

[Ye *et al.*, 2020] Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. *AAAI*, 34, 2020.

[Zhang *et al.*, 2020] Shaofeng Zhang, Meng Liu, and Junchi Yan. The diversified ensemble neural network. *NeurIPS*, 33, 2020.

[Zhou *et al.*, 2002] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.

[Zhou *et al.*, 2018] Tianyi Zhou, Shengjie Wang, and Jeff A Bilmes. Diverse ensemble evolution: Curriculum data-model marriage. In *NeurIPS*, pages 5909–5920, 2018.