DEEP ENSEMBLE POLICY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensemble learning, which can consistently improve the prediction performance in supervised learning, has drawn increasing attentions in reinforcement learning (RL). However, most related works focus on adopting ensemble methods in environment dynamics modeling and value function approximation, which are more essentially supervised learning tasks of the RL regime. Moreover, considering the inevitable difference between RL and supervised learning, the conclusions or theories of the existing ensemble supervised learning cannot be directly adopted to policy learning in RL. Adapting ensemble method to policy learning has not been well studied and still remains an open problem. In this work, we propose to learn the ensemble policies under the same RL objective in an end-to-end manner, in which sub-policy training and policy ensemble are combined organically and optimized simultaneously. We further theoretically prove that ensemble policy learning can improve exploration efficacy through increasing entropy of action distribution. In addition, we incorporate a regularization of diversity enhancement over the policy space which retains the ability of the ensemble policy to generalize to unseen states. The experimental results on two complex grid-world environments and one real-world application demonstrate that our proposed method achieves significantly higher sample efficiency and better policy generalization performance.

1 INTRODUCTION

In supervised learning, ensemble methods (Dietterich, 2000) such as bagging (Breiman, 1996) and boosting (Freund et al., 1996) have demonstrated great potential in a wide range of tasks like computer vision (Szegedy et al., 2015) and tabular data mining (Liu et al., 2020). For these ensemble learning methods, multiple base models are trained to perform the same task and their outputs are aggregated to generate a final decision, which has shown better performance than individual models (Lee et al., 2015). The reason behind the success of ensemble methods in supervised learning is that the aggregation in ensemble methods can reduce the variance of final predictions (Dietterich, 2000).

Considering the great success of ensemble methods in supervised learning, applications of ensemble learning in reinforcement learning have also be studied. They mainly conduct ensemble learning in environment dynamics modeling (Kurutach et al., 2018) and value function approximation (Anschel et al., 2017), which mainly inherits the advantages from the ensemble supervised learning to enhance the supervised function approximation in reinforcement learning. Nevertheless, these tasks are only part of the reinforcement learning regime. Moreover, the mechanisms behind the policy learning and supervised learning are of huge gap thus the ensemble learning theory from the supervised learning cannot directly interpret and guide adopting ensemble method in policy learning.

Incorporating ensemble methods in policy learning has not been well studied in the existing works and still remains an open problem. Some works follow the ensemble supervised learning and achieve diversity of sub-policies through a trivial approach of *individually* training various policies and simply aggregating them ex post facto (Wiering & Van Hasselt, 2008; Duell & Udluft, 2013), which has few guarantees to improve the ensemble performance since they have not considered policy ensemble and policy learning as a whole optimization problem. The other works incorporate divide-and-conquer principle through either initial state clustering (Ghosh et al., 2017) or information theory based reward decomposition (Goyal et al., 2019) to divide the state space, and derive a set of specialized policies accordingly. But the sub-policies are not aware of the whole state space, which may significantly hurt the performance especially in *deep* reinforcement learning scenario and result

in poor ensemble effectiveness. Moreover, whether and how ensemble method would benefit policy optimization still remain unsolved. It also requires additional attention on the training strategy of sub-policies within the ensemble method.

In this paper, we rigorously treat ensemble policy learning as a first class problem to explicitly address the following questions including: i) what is the reasonable yet effective policy ensemble strategy in deep reinforcement learning and ii) how it helps to improve common policy learning.

We resolve the ensemble policy learning from two aspects. On one side, we argue that ensemble learning and policy learning should be considered as an organic whole system. We combine sub-policy training and decision aggregation as a whole and optimize them under the unified ensemble objective. Specifically, multiple sub-policies are optimized with data collected by the ensemble policy which aggregates all the co-training sub-policies for final decision. And we theoretically prove that the decision aggregation of co-training sub-policies helps in efficient exploration thus significantly improves the sample efficiency. To guarantee the ensemble performance, an ensemble-aware loss is introduced to encourage the cooperation among sub-policies. Considering that the induced diversity enhancement can essentially improve the ensemble performance in ensemble supervised learning (Rame & Cord, 2021; Zhou et al., 2018; Zhang et al., 2020), on the other side, we incorporate diversity strengthen regularization within the policy space to further improve the ensemble performance. We empirically found that it can improve policy generalization in real-world application because the diversity strengthen 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 this paper are threefold and listed as below.

- 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, this paper is the first work that introduces the diversity strengthen regularization on policy space for ensemble policy learning.
- Demonstrated by the experiments on two complex grid-world environments and one real-world application, our proposed ensemble policy learning method yields significantly better sample efficiency and the diversity strengthen regularization also provides a promising solution for improving policy generalization.

2 Related Work

Ensemble Supervised Learning In supervised learning, ensemble method has become a simple yet powerful approach to improve the prediction performance by aggregating the outputs from multiple base models (Zhou et al., 2002; Moghimi et al., 2016; Zhou et al., 2018; Rame & Cord, 2021). Diversity among different base models is a key reason for the success of ensemble methods in supervised learning which can reduce the variance of the aggregated predictions thus improve generalization (Dietterich, 2000). Moreover, another justification on the success of ensemble methods indicates that the aggregation operation enriches the space of the hypotheses considered by base models thus results in better performance (Domingos, 1997). However, due to the essential difference between supervised learning and reinforcement learning, the observation and theory of ensemble methods in supervised learning may not hold in the latter scenario.

Ensemble Methods in Reinforcement Learning The recent works applying ensemble methods in reinforcement learning are mainly focusing on environment dynamics modeling and value function approximation. For environment dynamics modeling, Chua et al. (2018) and Janner et al. (2019) train a set of models in parallel and reduce the model variance by randomly sampling a dynamics model at each step during rollout. Kurutach et al. (2018) explicitly derive several environment models to stabilize the model-based policy learning. As for value function approximation, Q-function ensemble has been widely used (Haarnoja et al., 2018; Kumar et al., 2019; Anschel et al., 2017) to alleviate the over-estimation of the ground truth value. Q-function ensemble is also a useful approach to measure the uncertainty of the given (state, action) pairs, thus it can help in exploration (Lee et al., 2021) or 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 supervised learning and it has a huge gap with policy learning in reinforcement learning. Among the existing works on ensemble policy learning, some works simply aggregate individually trained policies ex post facto which has no guarantee to improve the ensemble performance. To generate a set of sub-policies, differently initialized parameters (Faußer & Schwenker, 2015; Duell & Udluft, 2013) or different reinforcement learning algorithms (Wiering & Van Hasselt, 2008) are used. However, we find that ensemble individually trained policies contributes little to the final ensemble performance. The other works incorporate divide-and-conquer principle to divide the state space and derive a set of specialized polices accordingly, and then they finally aggregate these specialized policies together to solve the original task. DnC (Ghosh et al., 2017) divides the whole task into several sub-tasks based on the clustering of the initial states, then it derives several policies accordingly and periodically distills the knowledge into a center policy. However, in many environments, initial states cannot provide enough information for reasonable state space segmentation, which makes this method fail in these scenarios. Based on information theory, Goyal et al. (2019) design a competition mechanism among policies and adopt reward decomposition to train several diverse policies with decorrelation loss. Due to the division of the state space, the sub-policy is not aware of the whole state space, which may significantly hurt the performance especially in *deep* reinforcement learning scenario and result in poor ensemble effectiveness.

Diversity Enhancement In supervised learning, it is acknowledged that the diversity among base models can reduce the variance and improve the performance (Domingos, 1997; Zhang et al., 2020). In reinforcement learning, diversity enhancement is mainly used to derive a set of diverse polices. In population-based reinforcement learning, diversity enhancement has been achieved through minimizing KL divergence, maximum mean discrepancy or maximizing the determinant of the kernel matrix in determinantal point process for efficient exploration (Hong et al., 2018; Masood & Doshi-Velez, 2019; Parker-Holder et al., 2020). It has also been considered in unsupervised skill discovery (Eysenbach et al., 2018; Sharma et al., 2019). But they only pursue high diversity without considering the actual rewards of the environment, which has been found not useful in our experiments. And Goyal et al. (2019) propose a diversity enhancement regularization on the representation space in ensemble policy learning. In contrast, we borrow the idea from population-based reinforcement learning and impose the diversity strengthen regularization on the policy space to further improve generalization of the ensemble policy.

3 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, also called the state transition probability of the environment given state s and action a. And $\Omega(\mathcal{S})$ is the set of distributions over $\mathcal{S}$. The initial state s_0 of the environment follows $p_0 : S \mapsto \mathbb{R}$ which is the distribution of the initial state s_0. $r(s, a) : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function. $\eta(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{T} r(s_t, a_t) \right]$ denotes the expectation of the cumulative reward that a policy π obtained by interacting with the environment within a trajectory $\tau = \{(s_t, a_t)\}_{t=0}^{T}$. And the objective of reinforcement learning is to maximize the reward expectation $\eta(\pi_\theta)$ as

$$\arg\max_{\theta} \mathbb{E}_{\pi_\theta} \left[\sum_{t=0}^{T} r(s_t, a_t) \right] , \tag{1}$$

where θ is the parameter of the policy π.

4 DEEP ENSEMBLE POLICY LEARNING

In this section, we present our deep ensemble policy learning algorithm. We first motivate our design in the framework, and then introduce the details of a practical deep ensemble policy learning algorithm. The overview of the architecture is illustrated in Figure 1 and a detailed training procedure is shown in Algorithm 1.

4.1 POLICY ENSEMBLE

In this section, we discuss the motivation for the design of policy ensemble in our framework on the training of sub-policies and data collection.

As has been widely used in ensemble supervised learning (Zhou et al., 2002; Lee et al., 2015), value-based RL methods (Kumar et al., 2019; Anschel et al., 2017) and environment modeling (Kurutach et al., 2018), the approximated function has been aggregated by a set of base components each of which can be optimized and work individually in the target task. We first consider to maintain K sub-policies $\{\pi_{\theta_1}, \ldots, \pi_{\theta_k}, \ldots, \pi_{\theta_K}\}$ in parallel in our framework where $k \in [1, K]$ represents the index of sub-policy parameters. For better illustration, we denote π_k to represent each sub-policy parameterized by θ_k.

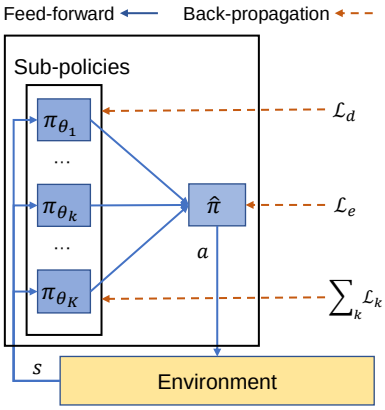


Figure 1: Framework of deep ensemble policy learning.

As is shown in Figure 1, we derive the ensemble policy $\hat{\pi}$ through aggregating the outputs from the sub-policies as

$$\hat{\pi}(\cdot|s) = \frac{1}{K} \sum_{k=1}^{K} \pi_k(\cdot|s; \theta_k) \,. \tag{2}$$

Receiving the state s from the environment, the sub-policies $\{\pi_k\}_{k=1}^K$ will produce their corresponding action distributions. After that, the overall action distribution of the ensemble policy $\hat{\pi}(\cdot|s)$ is calculated as the arithmetic mean of the action distribution $\pi_k(\cdot|s)$ derived from the sub-policies $\{\pi_k\}_{k=1}^K$, and the operation is named mean aggregation for short. Note that the ensemble policy $\hat{\pi}$ is parameter free and the whole set of parameters are $\{\theta_k\}_{k=1}^K$ of the sub-policies.

Finally, the action a will be taken from the overall action distribution as $a \sim \hat{\pi}(\cdot|s)$ and sent back to the environment. Note that, only the ensemble policy $\hat{\pi}$ is used to interact with the environment for collecting trajectories $\{\tau_i\}_{i \leq N}$ and these trajectories are used to update the sub-policies.

Discussion Sample efficiency is a key problem in RL scenario especially in real-world environment as the sampling cost of exploration is expensive and the advantages of our proposed deep ensemble policy learning should not come from more trajectories sampled by more sub-policies. Thus, we should *not* let the sub-policies interact with the environment directly. Taking all these factors into account, we use the ensemble policy $\hat{\pi}$ for exploration and training data collection in our framework and the data collected is shared by all the sub-policies. Moreover, in Theorem 1, we prove that the ensemble policy has significantly improved the exploration efficiency by simply aggregating the sub-policies, which leads to higher sample efficiency as shown in the experiments.

4.2 POLICY OPTIMIZATION

Though it has shown that simply aggregating the predictions of the base models in ensemble supervised learning leads to performance improvement (Zhou et al., 2018), the aggregation of the sub-policies does not provide any guarantees of promising ensemble performance. It is heuristically reasonable since RL is a sequential decision making problem which is more complex than supervised learning. Specifically, the aggregation of the outputs from individually trained sub-policies may derive undesirable action distribution and lead to bad states, especially in the case of the separately optimized sub-policies, such as Duell & Udluft (2013). It is necessary to take the optimization of the sub-policies as a whole system to enforce consistent learning paradigm for ensemble policy learning. Therefore, we first incorporate an ensemble-aware loss to ensure a good ensemble policy $\hat{\pi}$ and encourage the cooperation among the sub-policies.

We optimize the ensemble policy $\hat{\pi}$ to maximize the expected reward $\eta(\hat{\pi})$ as:

$$\arg\max_{\{\theta_k\}_{k=1}^K} \mathbb{E}_{\hat{\pi}} \left[\sum_{t=0}^{T} r(s_t, a_t) \right] \,. \tag{3}$$

Here we embed the ensemble policy learning into Proximal Policy Optimization (PPO) (Schulman et al., 2017) and derive the overall policy optimization loss with KL divergence as

$$\mathcal{L}_e(\hat{\pi}) = -\mathbb{E}_{\hat{\pi}'}\left[\sum_{t=0}^{T}\frac{\hat{\pi}(a_t|s_t)}{\hat{\pi}'(a_t|s_t)}\hat{A}_{\hat{\pi}'}(t)\right] + \mu\mathbb{E}_{\hat{\pi}'}\left[\sum_{t=0}^{T}\mathrm{KL}\left[\hat{\pi}'(\cdot|s_t), \hat{\pi}(\cdot|s_t)\right]\right], \tag{4}$$

where $\hat{\pi}'$ is the policy used for collecting the trajectory, μ is an adaptive penalty parameter to constrain the size of the policy update and $\hat{A}$ is the advantage function that can be implemented as a generalized advantage estimator (GAE) (Schulman et al., 2015). Note that we can also incorporate other policy optimization algorithms with policy ensemble since optimizing Eq. (3) is common. Taking Eq. (2) into Eq. (4), we can find that the policy gradient will be back-propagated through the policy ensemble procedure and each sub-policy will be updated simultaneously, which enforce the sub-policy training in a unified behavior under the same ensemble policy loss.

Motivated by the empirical results from ensemble supervised learning that the better base functions[1] lead to better ensemble performance, we think it is necessary to optimize the performance (i.e. $\eta(\pi_{\theta_k}), 1 \leq k \leq K$) w.r.t. these sub-policies. In order to directly improve the sub-policies through policy optimization, we can also adopt an individual training loss onto each sub-policy with the same trajectory collected by the ensemble policy $\hat{\pi}$ as:

$$\mathcal{L}_k(\pi_{\theta_k}) = -\mathbb{E}_{\hat{\pi}'}\left[\sum_{t=0}^{T}\frac{\pi_{\theta_k}(a_t|s_t)}{\hat{\pi}'(a_t|s_t)}\hat{A}_{\hat{\pi}'}(t)\right] + \mu\mathbb{E}_{\hat{\pi}'}\left[\sum_{t=0}^{T}\mathrm{KL}\left[\hat{\pi}'(\cdot|s_t), \pi_{\theta_k}(\cdot|s_t)\right]\right]. \tag{5}$$

Here the loss for each sub-policy π_{θ_k} directly optimizes θ_k through policy gradient by the data collected by ensemble policy $\hat{\pi}$, which does not impose additional interaction behaviors between the sub-policy and the environment thus keep the sample cost the same as that of a single policy.

There would be one potential risk of mode collapse in policy ensemble that all the sub-policies converge to one specific sub-policy and all the decisions would be taken from this specific policy. It is trivial since all the sub-policies share the similar training paradigm which makes these sub-policies may tend to behave similar. Therefore, a diversity strengthen regularization is necessary to prevent all the sub-policies from collapsing into a singular mode and to ensure the diversity among the sub-policies.

Recall that, in ensemble supervised learning, diversity has been shown to reduce the prediction variance (Zhou et al., 2002; 2018) and improve generalization. In population-based RL methods, diversity enhancement has been introduced for efficient exploration (Hong et al., 2018; Masood & Doshi-Velez, 2019; Parker-Holder et al., 2020). Therefore, we introduce a regularization to reinforce the diversity among different sub-policies for further improving ensemble performance. Intuitively, the diversity strengthen regularization encourages the action distributions proposed by different sub-policies to be orthogonal with each other. Specifically, we focus on the environments with discrete action space, the diversity strengthen regularization is defined as

$$\begin{aligned}\mathcal{L}_d &= \frac{2}{K(K-1)}\sum_{1\leq i\leq j\leq K}\langle\pi_i(\cdot|s), \pi_j(\cdot|s)\rangle, \\ \text{where } \langle\pi_i(\cdot|s), \pi_j(\cdot|s)\rangle &= \sum_a \pi_i(a|s)\pi_j(a|s).\end{aligned} \tag{6}$$

As for continuous control environment, a mean squared error among the sub-policies can serve as a diversity strengthen regularization.

Discussion In some related works of RL, diversity can be measured in state level (Hong et al., 2018) or trajectory level (Masood & Doshi-Velez, 2019). And we measure the diversity in state level because the sub-policy cannot directly collect trajectories. We adopt inner product as diversity measure because it is more computationally efficient (Li et al., 2012). And the diversity regularization in our method aims at getting a set of sub-policies which are well-behaved and mutually

[1] In supervised machine learning, the function is often named as "model" which can be approximated by a series of different methods such as deep neural network. To avoid misunderstanding to the environment model in RL, we use "function" here for better reference.

different, while Eysenbach et al. (2018) only pursue diverse sub-policies without maximizing the real reward of the environment, which has little guarantees to perform well considering the real reward as illustrated in the experiment.

In conclusion, the overall loss of our framework is defined as

$$\mathcal{L} = \mathcal{L}_e(\hat{\pi}) + \alpha \sum_{k=1}^{K} \mathcal{L}_k(\pi_{\theta_k}) + \beta \mathcal{L}_d, \tag{7}$$

where α and β are the hyper-parameters controlling the impact of the respective terms. We summarize the overall ensemble policy learning algorithm in Algorithm 1.

Algorithm 1: Deep Ensemble Policy Learning

Input: Training epoch number I, number of trajectories N collected per epoch, repeat times S after data collection

Initialize all the sub-policies $\{\pi_{\theta_k}\}_{k=1}^{K}$ with their corresponding parameters $\{\theta_k\}_{k=1}^{K}$.
for $i = 1$ *to* I **do**
  Collect N trajectories $\Gamma = \{\tau_i\}_{i \leq N}$ from the real environment using $\hat{\pi}$ defined as Eq. (2).
  for $s = 1$ *to* S **do**
    Update the sub-policies through $\hat{\pi}$ using Eq. (4) over Γ.
    Update the sub-policies directly using Eq. (5) over Γ.
    Enhance the diversity among the sub-policies by Eq. (6) over Γ.

4.3 Theoretical analysis

Theorem 1 (Mean aggregation encourages exploration)**.** *The entropy of the ensemble policy $\hat{\pi}$ is no less than the average entropy of the sub-policies, i.e., $\mathcal{H}(\hat{\pi}) \geq \frac{1}{K} \sum_{k=1}^{K} \mathcal{H}(\pi_{\theta_k})$.*

The proof of Theorem 1 can be found in Appendix A.1. Theorem 1 shows that the mean aggregation operation can lead to a behavior policy, i.e., the ensemble policy, with equal or larger entropy value than the average entropy of the sub-policies at any state, which illustrates more effective exploration in the policy learning procedure. Thus, through aggregating the sub-policies during training, $\hat{\pi}$ has guaranteed performance in exploration in the whole state space and improves the sample efficiency of policy learning. We have also observed the corresponding phenomenon in the experiments. It also reflects the necessity of incorporating the unified optimization for the policy ensemble, which has been discussed above.

5 Experiments

We focus our analysis on environments spanning two different domains, a grid-world environment named Minigrid (Chevalier-Boisvert et al., 2018) and a real-world application named financial algorithmic trading (Fang et al., 2021). The experiments and the analysis in this section are led by the following two research questions (RQs). **RQ1**: Does our proposed method achieve higher sample efficiency through policy ensemble? **RQ2**: Is the generalization performance of the proposed ensemble policy better than the other compared methods?

5.1 Compared methods

We compare our proposed method with its variants and the following baselines.

- **PPO** (Schulman et al., 2017) is a state-of-the-art policy optimization method and is used as the base learner in our method. So we use PPO as a standard policy learning baseline without ensemble for comparison.
- **PE** (Policy Ensemble) is based on a traditional policy ensemble method (Duell & Udluft, 2013) which trains K policies *individually* by PPO algorithm and then aggregates them to solve the task. In our paper, we consider two aggregation operation, 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. (2) respectively. And we denote these two methods as **PEMV** and **PEMA** for short.

- **DnC** (Ghosh et al., 2017) is based on divide-and-conquer principle which partitions the initial state space into K slices, and optimizes an ensemble of policies each on a different slice. During training, these policies are periodically distilled into a center policy that is utilized 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 by a well-designed competition mechanism based on information theory.
- **DEPL** is our proposed method described above, which has two other variants for ablation study: **DEPL-Div** is the method *without* diversity strengthen regularization defined in Eq. (6) and **DEPL-Ens** is the method *without* ensemble-aware loss defined in Eq. (4).

In all methods, we take PPO as the base policy optimization method. To ensure a fair comparison, all the ensemble methods have roughly the same number of parameters (i.e., K times the parameter size of PPO) and the number of trajectories collected in one epoch for training is kept the same for all the methods which provides fairness for comparison of sample efficiency. In the experiments, we keep the number of base policies $K = 4$ for all the ensemble methods as default and more implementation details can be found in Appendix D.

5.2 Evaluation: Minigrid

For comparison of sample efficiency of the different methods, we consider two partial-observable environments in Minigrid (Chevalier-Boisvert et al., 2018): distributional-shift and multi-room, as shown in Figure 2, where the agent (red triangle mark) aims at reaching the given target position (green square mark) and a nonzero reward is provided only if the target position is reached. The view horizon of agent is limited thus partial observable as marked by the half-transparent grey area as in Figure 2. Moreover, these environments are challenging because of introduction of stochasticity during reset procedure. 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. And due to the longer distance between the start position and the goal, multi-room is more difficult. As a result, effective exploration is required because of the sparse reward setup of the environments thus these environments are suitable for testing the sample efficiency of the compared methods.

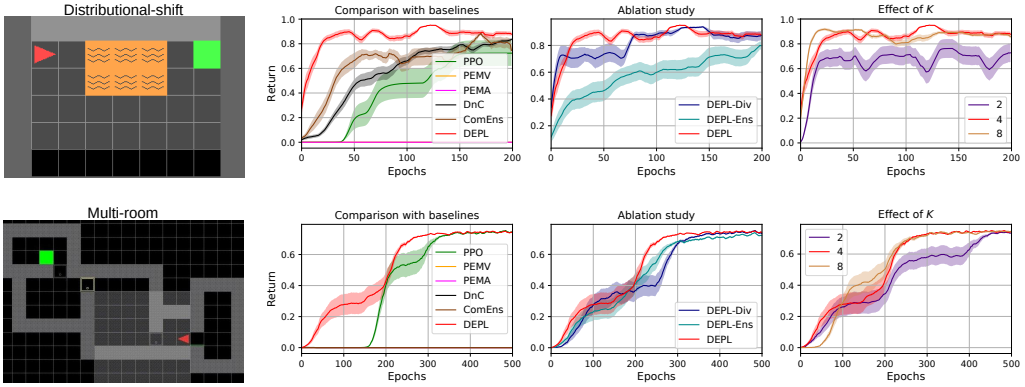


Figure 2: The evaluation results on Minigrid environments, all the results are conducted with five random seeds. The top and bottom rows show the information about distributional-shift and multi-room, respectively. **First column**: A snapshot of the environment where the red triangle and green square represent the position and the goal of the agent respectively. **Second column**: Learning curves of all the compared methods. **Third column**: Learning curves of DEPL and its variants. **Last column**: Learning curves of DEPL when K is set to different values.

As shown in Figure 2, DEPL enjoys the best sample efficiency and gets nonzero reward fastest among all the methods in both environments (**RQ1**). We notice that PEMA and PEMV fail (i.e., return $= 0$) in both environments within 200 epochs while PPO can get better performance. And we think that the failure can be attributed to two reasons. First, considering the number of samples needed for PPO to get a nonzero reward is large, PEMA and PEMV may need K times samples for

a nonzero reward because they individually train K PPO policies. Second, due to the asynchrony among the sub-policies of the traditional PE method, the useful knowledge can be overwhelmed thus neglected due to the aggregation operation by other worthless knowledge. The failure of PEMA and PEMV implies the necessity of an ensemble-award loss and explains the rationality of sampling with ensemble policy $\hat{\pi}$. Moreover, the failure of DnC and ComEns in multi-room environment, which is more challenging, can be attributed to the division operation on the state space that may hinder the ability on exploration to the full environment.

The ablation study in Figure 2 shows that DEPL outperforms its two variants, thus both of the diversity strengthen regularization and ensemble-aware loss appear to be crucial for the superior performance of DEPL. In addition, the returns of both DEPL 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 DEPL. 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. In majority voting, when $K = 2$ and sub-policies have different decision, ensemble policy cannot extract the valuable information from sub-policies thus cannot promote the performance or even get worse performance. And we think the similar reason leads to the poor performance in DEPL when $K = 2$ because of the similarity between majority voting and mean aggregation.

5.3 Evaluation: Financial Algorithmic Trading

The requirement of generalization is widespread in the real-world tasks, but the problem is not well studied and still remains an important and difficult task for RL scenario (Cobbe et al., 2019). To evaluate generalization ability of different methods, we conduct experiments on financial order execution (Fang et al., 2021) which is a fundamental yet challenging problem in algorithmic trading and there are many RL methods working on it (Ning et al., 2018; Lin & Beling, 2020; Fang et al., 2021). In order execution task, 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, within a time horizon such as one day.

In particular, the environments are usually formulated as training, validation and test phases each of which is corresponding to a specific time range within the data, that is quite different to the traditional RL tasks where the environment keeps the same during training and test. Specifically, *training environment* is used for data collection and policy optimization, *validation environment* is responsible for hyper-parameter selection. *Test environment* is unavailable for the agent during training and is used to measure the generalization ability of the learned policy trained with the selected best hyper-parameters.

Table 1: The dataset statistics of financial order execution 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

The task of order execution is challenging for policy optimization. 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 learned agent has to make decisions in unfamiliar states during testing, which puts forward a strong demand for the generalization of the learned policy. Therefore, the performance of the learned agent in test environment is a good surrogate evaluation of the generalization ability, which has been used widely in the related works (Lin & Beling, 2020; Fang et al., 2021). Moreover, the agent has to make decision based on the noisy and yet imperfect market information, which may induce the agent to make wrong decision and lead to the poor effectiveness of the agent. And due to the large number of stocks which fluctuate in different patterns, the agent must be able to handle various events for better trading performance, which also makes the task challenging.

Following Fang et al. (2021), the reward in order execution environment is composed of price advantage (PA) and market impact penalty where PA encourages the policy to get better profit than the average benchmark market price. The averaged PA and reward of the test orders in the test environment are taken as evaluation metrics in order execution task. We conduct the experiment 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 1.

The results of different methods are reported in Table 2. As expected, DEPL achieves the best performance in both PA and reward in two datasets, suggesting that our proposed method has great potential in generalizing to new 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. (4) which can encourage the coordination among sub-policies, is necessary. In addition, the performance degradation of DEPL-Ens also illustrates the importance of the ensemble-aware loss. Moreover, from the comparison between DEPL and DEPL-Div, we find that the diversity strengthen regularization further improves the generalization performance. This phenomenon coincides with the observations in supervised learning (Zhou et al., 2002) that the diversity among base models can reduce the variance, alleviate the over-fitting problem and improve the generalization performance of the ensemble method.

Table 2: Test performance on financial order execution environment; the higher metric value means the better performance. The results are the average of all test orders over ten random seeds.

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

To evaluate the effect of the number of sub-policies K, we conduct experiments when $K \in \{2, 4, 8\}$ in dataset 1801-1908 and the results are shown in Table 3. Similar with 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 result in a better performance, but $K = 4$ seems to be a good choice based on our experimental results in Minigrid and order execution.

Table 3: Results on different K

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

5.4 Diversity Analysis

After analyzing the performance of diversity strengthen regularization in improving the policy generalization, we further verify whether the regularization diversifies the sub-policies. Motivated by the metric applied as diversity measure in continuous action space (Hong et al., 2018), here we consider the action disagreement (AD) to measure the diversity among sub-policies as $\text{AD} = \frac{2}{K(K-1)} \sum_{1 \le i \le j \le K} \frac{1}{|M|} \sum_{s \in M} \mathbb{I}(\arg\max_a \pi_i(a|s) \neq \arg\max_a \pi_j(a|s))$, where M is a set of states. In dataset 1801-1908, the AD values of DEPL and DEPL-Div are 15.9% and 14.3%, respectively, which proves that the regularization term improves the diversity among sub-policies.

Motivated by the above results, we want to explore whether pure diversity can lead to good generalization performance. So we train **Diayn** (Eysenbach et al., 2018) policies for unsupervised skill discovery in dataset 1801-1908 of order execution environment and then aggregate them. However, we find Diayn even cannot get positive PA or reward (i.e., -0.99 and -8.37, respectively). Thus, we can derive two conclusions. First, a purely diversity-oriented method e.g., Eysenbach et al. (2018), is not promising in ensemble policy learning considering real rewards. Second, the diversity should serve as an inductive bias and be incorporated into the ensemble policy learning as a regularization as that in Eq. (7) of DEPL. More analysis on diversity can be found in Appendix B.

6 Conclusion and Future Work

In this paper, we focus on ensemble policy learning and propose an end-to-end ensemble policy optimization framework combining sub-policy training and policy ensemble as a whole. The base policies are updated simultaneously under the ensemble loss and a diversity enhancement regularization. We provide a theoretical analysis of how ensemble works in policy learning. The experiments on two complex grid-world environments and a challenging real-world task demonstrate DEPL substantially outperforms the baselines in sample efficiency and policy generalization performance. In the future work, we plan to incorporate more flexible sub-policy ensemble with dynamic aggregation weights for different sub-policies and derive more theoretical analysis of ensemble policy learning.

ETHICS STATEMENT

Ensemble policy learning has not been well studied in reinforcement learning. To the best of our knowledge, there are no intentional ethical issues created by the proposed method.

REPRODUCIBILITY STATEMENT

The datasets used in this paper are all publicly available, and all implementation details are introduced either in the main paper or in the appendix. In addition, we will release the source code of all the experiments upon paper acceptance.

REFERENCES

Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International conference on machine learning*, pp. 176–185. PMLR, 2017.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. `https://github.com/maximecb/gym-minigrid`, 2018.

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

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 1282–1289. PMLR, 2019.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Pedro M Domingos. Why does bagging work? a bayesian account and its implications. In *KDD*, pp. 155–158. Citeseer, 1997.

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

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

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.

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

Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pp. 148–156. Citeseer, 1996.

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

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.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 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.

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.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *NeurIPS*, pp. 12519–12530, 2019.

Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*, 2019.

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

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6131–6141. PMLR, 2021.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Nan Li, Yang Yu, and Zhi-Hua Zhou. Diversity regularized ensemble pruning. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 330–345. Springer, 2012.

Siyu Lin and Peter A Beling. An end-to-end optimal trade execution framework based on proximal policy optimization. In *IJCAI*, pp. 4548–4554, 2020.

Zhining Liu, Wei Cao, Zhifeng Gao, Jiang Bian, Hechang Chen, Yi Chang, and Tie-Yan Liu. Self-paced ensemble for highly imbalanced massive data classification. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 841–852. IEEE, 2020.

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.

Mohammad Moghimi, Serge J Belongie, Mohammad J Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. Boosted convolutional neural networks. In *BMVC*, volume 5, pp. 6, 2016.

Brian Ning, Franco Ho Ting Lin, and Sebastian Jaimungal. Double deep q-learning for optimal execution. *arXiv preprint arXiv:1812.06600*, 2018.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, and Stephen Roberts. Effective diversity in population based reinforcement learning. *arXiv preprint arXiv:2002.00632*, 2020.

Alexandre Rame and Matthieu Cord. Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. *arXiv preprint arXiv:2101.05544*, 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.

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.

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

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

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.

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.

Shaofeng Zhang, Meng Liu, and Junchi Yan. The diversified ensemble neural network. *Advances in Neural Information Processing Systems*, 33, 2020.

Tianyi Zhou, Shengjie Wang, and Jeff A Bilmes. Diverse ensemble evolution: Curriculum data-model marriage. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 5909–5920, 2018.

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.

A PROOF

A.1 PROOF OF THEOREM 1

Theorem 1 (Mean aggregation encourages exploration)**.** *The entropy of the ensemble policy $\hat{\pi}$ is no less than the average entropy of the sub-policies, i.e., $\mathcal{H}(\hat{\pi}) \geq \frac{1}{K}\sum_{k=1}^{K}\mathcal{H}(\pi_{\theta_k})$.*

Proof.

$$
\begin{aligned}
H(\hat{\pi}) - \frac{1}{K}\sum_{k=1}^{K}\mathcal{H}(\pi_{\theta_k}) &= -\sum_{a}\frac{\sum_{k=1}^{K}\pi_{\theta_k}(a|s)}{K} log\left(\frac{\sum_{k'=1}^{K}\pi_{\theta_{k'}}(a|s)}{K}\right) \\
&\quad + \frac{1}{K}\sum_{k=1}^{K}\sum_{a}\pi_{\theta_k}(a|s)log(\pi_{\theta_k}(a|s)) \\
&= -\frac{1}{K}\sum_{k=1}^{K}\sum_{a}\pi_{\theta_k}(a|s)\left(log\left(\frac{\sum_{k'=1}^{K}\pi_{\theta_{k'}}(a|s)}{K}\right) - log(\pi_{\theta_k}(a|s))\right) \\
&= -\frac{1}{K}\sum_{k=1}^{K}\sum_{a}\pi_{\theta_k}(a|s)log\left(\frac{\sum_{k'=1}^{K}\pi_{\theta_{k'}}(a|s)}{K\pi_{\theta_k}(a|s)}\right) \\
&\geq -\frac{1}{K}\sum_{k=1}^{K}log\left(\sum_{a}\pi_{\theta_k}(a|s)\frac{\sum_{k'=1}^{K}\pi_{\theta_{k'}}(a|s)}{K\pi_{\theta_k}(a|s)}\right) \text{ (Jensen's inequality)} \\
&= -\frac{1}{K}\sum_{k=1}^{K}log\left(\frac{1}{K}\sum_{a}\sum_{k'=1}^{K}\pi_{\theta_{k'}}(a|s)\right) \\
&= -\frac{1}{K}\sum_{k=1}^{K}log(1) = 0.
\end{aligned}
$$

□

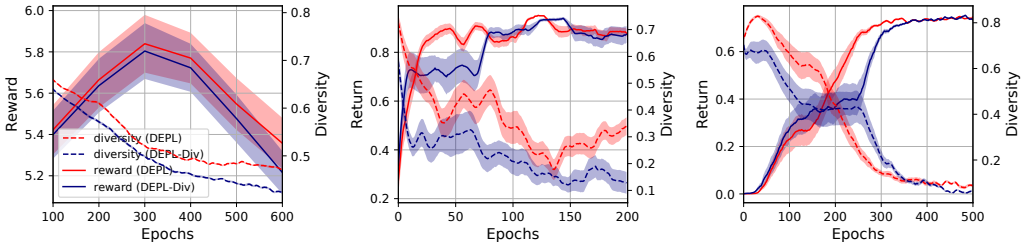
B FURTHER ANALYSIS OF DIVERSITY

In this section, we plot the learning curves about diversity and performance of DEPL and DEPL-Div, i.e., DEPL without regularization of diversity enhancement, on the 1801-1908 dataset of order execution, distributional-shift and multi-room in Figure 3.

For order execution, to show the effect of diversity strengthen regularization on policy generalization performance, we test the reward of DEPL and DEPL-Div on the *test environment* in dataset 1801-1908 every 100 epochs and draw the curves on Figure 3(a). Because these tested policies are not selected by *validation environment*, the results in Figure 3(a) cannot exactly match Table 2. From Figure 3(a), we notice that DEPL has larger diversity and better reward compared with DEPL-Div, suggesting that the diversity strengthen regularization can improve the policy generalization. Moreover, we find that the diversity continues to decline during training process for both methods. And we think this phenomenon is reasonable based on the following two reasons. Firstly, all the sub-policies are trained with the same data, thus they tend to becomes similar during training. Secondly, from the results of Diayn shown in Section 5.4, we find that pure diversity is unhelpful in ensemble policy learning, thus the diversity decrease is not absolutely harmful and the phenomenon is reasonable.

After observing that diversity strengthen regularization improves policy generalization performance, we further explore its effect on sample efficiency. From the results of two Minigrid environments as shown in Figure 3, we observe that DEPL-Div outperforms DEPL at first in both environments, then the performance of DEPL increases rapidly with diversity declining, and finally DEPL firstly converges and gets better sample efficiency. We think that the observation can be attributed to the variety of sub-policies induced by diversity regularization. At first, the larger diversity of DEPL

encourages it to explore more, which results in the worse performance and a rapid increase in performance afterwards. And then, the diversity converges or fluctuates within a small range when the policy converges. In conclusion, diversity strengthen regularization can encourage exploration at the early period on the training process while converging along with the policy optimization.

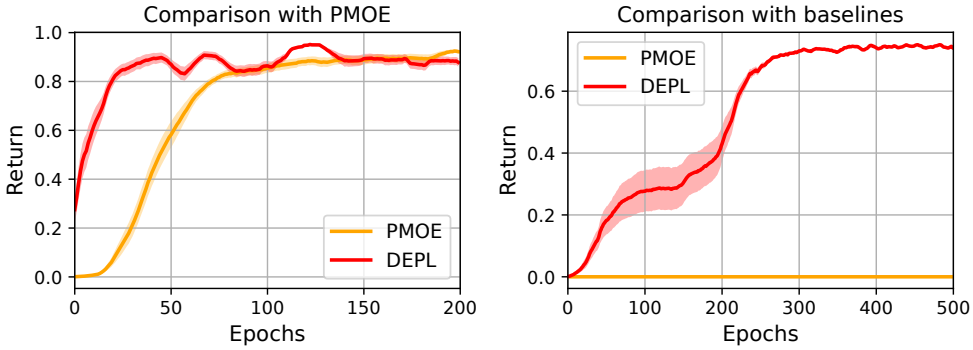


(a) Financial order execution (b) Distributional-shift (c) Multi-room

Figure 3: Learning curves about diversity and performance of DEPL and DEPL-Div

C Additional Baseline on Minigrid

Considering that DEPL is a special case under the paradigm of mixture of experts (Jacobs et al., 1991), thus we take PMOE (Ren et al., 2021), where a routing function is learned to weight the output of each expert, as a baseline. We compare PMOE with DEPL in Minigrid and the learning curves of both the methods are shown in Figure 4.



(a) Distributional-shift (b) Multi-room

Figure 4: Learning curves of DEPL and PMOE

From the results, we observe that DEPL consistently achieves better sample efficiency than PMOE, which can be attributed to two reasons. Firstly, the mean aggregation can promote the exploration while the weight generated by the routing function hurt the attribute. Secondly, to achieve diversity among sub-policies, a sample is only assigned to a single policy for policy optimization in PMOE, which can lead to a lower sample efficiency compared with DEPL where a diversity strengthen regularization is used to realize diversity.

D Implementation Details

All the methods in our paper are implemented with PyTorch 1.7. For PPO, we apply the implementation of tianshou[2]. For DnC[3] and Diayn[4], we adopt the open-source implementations from their paper in our experiments. And we implement ComEns since it has not been open-sourced. Considering the training is more stable on the environments of Minigrid than that on financial order

[2]https://github.com/thu-ml/tianshou
[3]https://github.com/dibyaghosh/dnc
[4]https://github.com/ben-eysenbach/sac

execution, thus we respectively conduct the evaluation of every compared method with 5 and 10 random seeds.

For Minigrid, the sub-policy for all ensemble methods except for ComEns is a convolutional network and the architecture of the networks is the same. Due to the special design of ComEns, its subpolicy needs one more Multi-Layer Perceptron (MLP) with 64 units to get a latent distribution for information amount computation compared with others. During each training epoch, 100 trajectories are collected (i.e., $N = 100$). The number of epochs (i.e., I) for distributional-shift and multi-room are 200 and 500, respectively.

Following Fang et al. (2021), a recurrent neural network is used as the architecture of the subpolicy for all ensemble methods except for ComEns. And ComEns still needs one more MLP with 64 units due to its special design. In this scenario, 10,000 trajectories are collected per epoch (i.e., $N = 10,000$) and the number of total training epochs is 600 (i.e., $I = 600$). During training process, the learned policy is evaluated in the validation environment every 50 epochs, and the policy with the best validation performance is selected as the final policy. Then, in test environment, the mean value of PA and reward metrics are calculated over the whole test dataset as the final performance in financial order execution environment.

For both Minigrid and order execution scenarios, the collected trajectories are used for 4 times during an epoch (i.e., $S = 4$).

As for the hyperparameters, all the trainable parameters are optimized by Adam optimizer with learning rate selected from $\{0.0001, 0.0003\}$ for all the compared methods. For our method, the α and β hyperparameters defined in Eq. (7) are selected from $\{0.5, 1.0\}$ and $\{0.01, 0.025, 0.04\}$, respectively. For ComEns, the coefficients in the loss function is selected from $\{0.001, 0.005, 0.009\}$ as recommended in Goyal et al. (2019). For DnC, considering a fixed penalty might yield different magnitudes of constraint for each environment, the coefficient of KL penalty is searched within $\{0.001, 0.01, 0.1, 1\}$ on each task.

All the experiments are run on a single NVIDIA P100 GPU with an Intel(R) Xeon(R) Platinum 8171M CPU. We can complete each experiment within three hours on distributional-shift and multi-room environments and three days on financial order execution scenario.