Exploring Parameter Space with Structured Noise for Meta-Reinforcement Learning

Hui Xu[1*], **Chong Zhang**[2], **Jiaxing Wang**[3,4], **Deqiang Ouyang**[1], **Yu Zheng**[2] and **Jie Shao**[1,5†]
[1]University of Electronic Science and Technology of China
[2]Tencent Robotics X
[3]Institute of Automation, Chinese Academy of Sciences
[4]University of Chinese Academy of Sciences
[5]Sichuan Artificial Intelligence Research Institute
{hui_xu, ouyangdeqiang}@std.uestc.edu.cn, aerentzhang@gmail.com,
jiaxing.wang@nlpr.ia.ac.cn, petezheng@tencent.com, shaojie@uestc.edu.cn

Abstract

Efficient exploration is a major challenge in Reinforcement Learning (RL) and has been studied extensively. However, for a new task existing methods explore either by taking actions that maximize task agnostic objectives (such as information gain) or applying a simple dithering strategy (such as noise injection), which might not be effective enough. In this paper, we investigate whether previous learning experiences can be leveraged to guide exploration of current new task. To this end, we propose a novel Exploration with Structured Noise in Parameter Space (ESNPS) approach. ESNPS utilizes meta-learning and directly uses meta-policy parameters, which contain prior knowledge, as structured noises to perturb the base model for effective exploration in new tasks. Experimental results on four groups of tasks: cheetah velocity, cheetah direction, ant velocity and ant direction demonstrate the superiority of ESNPS against a number of competitive baselines.

1 Introduction

Reinforcement Learning (RL) has achieved great success in various applications. In RL, the agent improves its future rewards by exploring insufficiently understood states and actions so that preventing premature convergence. Prior works have proposed a wealth of strategies, e.g., exploration by injecting random noise in the action space [Sutton and Barto, 2018], following task-agnostic criteria such as state visitation counts [Bellemare *et al.*, 2016; Kearns and Singh, 2002; Tang *et al.*, 2017], information gain [Houthooft *et al.*, 2016] and curiosity [Sun *et al.*, 2011; Pathak *et al.*, 2017], and exploration by sampling and ensemble of models [Osband *et al.*, 2016]. These techniques are task agnostic and do not contain any prior information on the way to better explore the task.

In practice, agents are often supposed to be able to handle distinct but related tasks. This leads to meta-RL. To utilize previous learning experiences. Model Agnostic Meta Learning (MAML) [Finn *et al.*, 2017] propose to learn a meta-policy which contains common structures of various tasks. The agent can then adapt agilely to a new task from the learned meta-policy. Meta learning decouples the common structures and task-specific information of the tasks so that it can be utilized to acquire efficient and guided exploration for a specific task. Gupta *et al.* [2018] proposed Model Agnostic Exploration with Structured Noise (MAESN), which learns a task-specific latent variable z. $z \sim \mathcal{N}(\mu, \sigma^2)$ containing the task-specific knowledge, is then concatenated to the states and used as structured prior noise for efficient and guided exploration. MAESN obtains an impressive exploration behavior in sparse reward tasks. However, computing dataset level features is always difficult and MAESN does not yield a better result than the original MAML in dense reward cases.

Inspired by MAESN, we propose a novel Exploration with Structured Noise in Parameter Space (ESNPS) approach by injecting structured noise directly in the policy parameter space. Different from MAESN which uses the task-specific latent variable as noise, we show that meta-policies trained on different partitions of meta-train tasks, containing diverse prior knowledge, can be used as additive structured noise up to a scaling factor. Besides the bonuses of prior guided exploration, training with parameter noise in high-dimensional neural network parameter space [Plappert *et al.*, 2018; Fortunato *et al.*, 2018] helps to escape local optima [Jin *et al.*, 2017] in complex non-convex problems. To ensure consistency in actions and temporally coherent stochasticity, the policy network is perturbed only at the beginning of an episode.

Traditional parameter space noise injection techniques, however, are not directly applicable for meta-RL. In Plappert *et al.* [2018] and Fortunato *et al.* [2018], policy parameters are perturbed and updated with reparameterization trick [Kingma and Welling, 2014]. This optimization requires substantial gradient steps to avoid unpredictable high variance, but Meta-RL enjoys fast adaptation and converges in a few

[*]This work was primarily done during the author's internship at Tencent Robotics X.

[†]Contact Author: Jie Shao.

gradient steps on a new task. Instead of optimizing the original policy parameters, we directly adapt the perturbed policy parameters. Among the fine-tuned models perturbed by different structured noise containing different prior knowledge, the best performing one is picked.

Extensive experiments are conducted on cheetah velocity, cheetah direction, ant velocity and ant direction. The results show that our proposed ESNPS effectively explores testing tasks, outperforming a set of competitive baselines. The main contributions of our work are as follows:

- We propose ESNPS, which effectively utilizes previous search experiences as structured noise for directed efficient exploration in a new task. Specifically, to the best of our knowledge, this is the first time to use the meta parameters of policy networks as parameter space structured noise.
- We consider the high variance problem caused by injecting noise in parameter space in meta-RL and propose a new optimization strategy suitable for fast adaptation in meta-learning.
- Extensive experiments demonstrate the effectiveness of the constructed structured noise in directed exploration for a specific task as well as the superiority of our proposed method against a set of competitive baselines.

2 Related Work

A crucial problem of reinforcement learning is how to explore effectively, and prior works have made great efforts for it. State visitation methods [Kearns and Singh, 2002; Brafman and Tennenholtz, 2001] are based on optimism in the face of uncertainty, which is yet constrained by small state-action space. The concept of curiosity [Sun *et al.*, 2011] is beneficial to the exploration. The information gain [Houthooft *et al.*, 2016] and parameter space exploration [Plappert *et al.*, 2018; Fortunato *et al.*, 2018] are also used for exploration strategies. Additionally, Plappert *et al.* [2018] and Fortunato *et al.* [2018] use noise to perturb the policy network in parameter space. This ensures consistency in actions and temporally coherent stochasticity. However, these exploration strategies are largely task-agnostic, in that they aim to explore without exploiting the particular structure of the tasks. This approach using the samples from the perturbed policy to update the non-perturbed network is not suitable for fast adaptive meta-learning method. In this paper, we investigate whether previous learning experiences can be leveraged to guide exploration of the current new task. This is achieved by incorporating the idea of meta-learning.

Meta-learning is also known as learning-to-learn [Mitchell and Thrun, 1992; Vilalta and Drissi, 2002; Lemke *et al.*, 2015], where a model extracts common structures of related but distinguished tasks so that a similar and previously unseen task can be efficiently solved with the prior knowledge. Various meta-learning methods have been proposed [Hochreiter *et al.*, 2001; Vinyals *et al.*, 2016; Mishra *et al.*, 2017] and the most relevant is the Model-Agnostic Meta-Learning (MAML) [Finn *et al.*, 2017]. MAML and its extensions [Finn *et al.*, 2018; Yoon *et al.*, 2018; Li *et al.*, 2017; Grant *et al.*, 2018; Nichol *et al.*, 2018] learn a set of meta-weights as initialization from where a few gradient steps yield good performance on a previously unseen task. Meta-learning decouples the common structures and task-specific information of the tasks so it can naturally be utilized to acquire efficient and guided exploration for a specific task. Closer to ours is MAESN [Gupta *et al.*, 2018], which utilizes MAML and additionally learns a latent variable $z \sim \mathcal{N}(\mu, \sigma^2)$, containing the task-specific knowledge as a structured noise for exploration. MAESN obtains an impressive exploration behavior in sparse reward tasks. However, in dense reward cases, MAESN does not yield better results than vanilla MAML. Compared with MAESN, Our ESNPS does not explicitly compute the task representation z, which is always difficult. Besides, as a parameter space exploration method, ESNPS enjoys the bonus of training with noised parameters, which helps to escape local optima.

3 Preliminaries

Suppose that a number of tasks $\{\mathcal{T}\}$ are sampled from a task distribution $p(\mathcal{T})$. Each task $\mathcal{T}_i$ is a Markov Decision Process (MDP) and is denoted as $\mathcal{T}_i = \{\mathcal{S}, \mathcal{A}, \mathcal{P}_i, \mathcal{R}_i, \gamma, \rho_i\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}_i : p_i\left(s_{t+1}|s_t, a_t\right)$ is a transition probability distribution, $\mathcal{R}_i : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is the reward function, ρ_i the initial state distribution and $\gamma \in (0, 1)$ is the discount factor. Policy parametrized by θ is denoted as π_θ. For each task $\mathcal{T}_i$, we collect a trajectory $\tau_{ij} = \{s_t, a_t, r_t\}_{t=0}^{H-1}$ where H is the actual trajectory horizon for the task, and the trajectory discounted return of the task $R(\tau_{ij}) = \sum_{t=0}^{H-1} \gamma r_t$. Now, the likelihood of a trajectory induced by policy π_θ is given by $p(\tau_{ij}|\theta) = \rho_i\left(s_0\right) \prod_{t=0}^{H-1} p_i\left(s_{t+1}|s_t, a_t\right) \pi_\theta\left(a_t|s_t\right)$. Meta-RL considers how to utilize previous learning experiences for efficient search on a new task. To find a procedure that can generate a good policy to adapt to the new task $\mathcal{T}_i$ though a few gradient steps, Finn *et al.* [2017] proposed Model Agnostic Meta-Learning (MAML), which maintains a meta-policy π_θ, from where a few gradient descent steps can lead to a significant increase of performance on previously unseen tasks $\mathcal{T}_i$. The fast task adaptation is performed by:

$$\phi_i = \theta + \beta \mathbb{E}_{\pi_\theta}\left[\sum_t R_i\left(s_t\right) \nabla_\theta \log \pi_\theta\left(a_t|s_t, p_{\mathcal{T}_i}\right)\right], \quad (1)$$

where β is the adaptation learning rate of policy and R is the trajectory discounted reward. The meta policy is updated by solving the following problem:

$$\max_\theta \sum_{\mathcal{T}_i} \mathbb{E}_{\pi_{\phi_i}}\left[\sum_t R_i\left(s_t\right)\right]. \quad (2)$$

The model learns the meta-policy π_θ as good initialization for adaptation when Equation (2) converges. By solving Equation (2) and getting the meta-policy, the agent can reach high reward with limited examples on a previously unseen task.

4 Our Method

To better explore a new task in the meta-RL setting, we propose Exploration with Structured Noise in Parameter Space

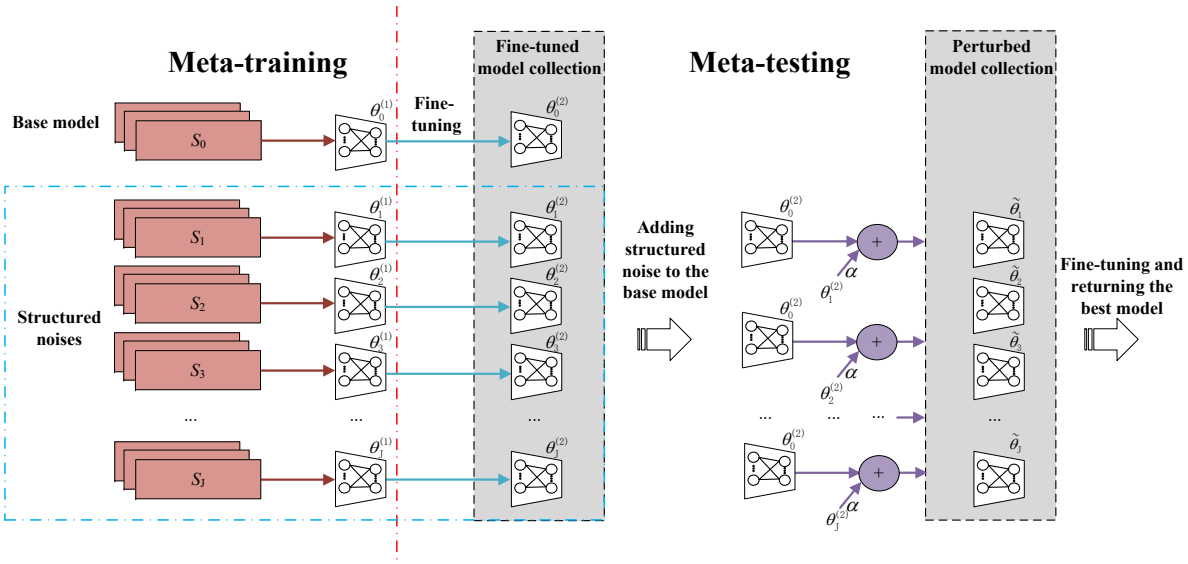

Figure 1: The framework of ESNPS has two phases: meta-training and meta-testing. In meta-training, we split the task set into $J+1$ partitions, and then train $J+1$ MAML models on these partitions. In meta-testing, we first fine-tune the $J+1$ models on the specific task, and put them into the fine-tuned model collection. The number of gradient steps k for fine-tuning is always set to 4. We set model θ_0 trained on the whole training set $\mathcal{S}_0 = \mathcal{T}$ as the base model, and other models as structured noises. Then, we iteratively perturb the base model θ_0 to generate perturbed models. Finally, we fine-tune each perturbed model $\tilde{\theta}_j, (j \in 1, 2, \cdots, J)$, and pick the best performing model as an output.

(ESNPS), whose framework is presented in Figure 1. ESNPS uses meta-policies trained on different partitions of meta-train, containing diverse prior knowledge, as additive structured noise for better exploration of a new task in the meta-testing phase. Among the perturbed models, after fine-tuning, the best performing one is picked. In the following, we first analyze how to apply parameter space noise without causing high variance problem in meta-RL setting. Then, we detail ESNPS in how to construct effective structured noise with meta-policies trained on different partitions of train tasks and discuss the intuition behind it. Finally, we introduce an adaptive scaling module to automatically determine the noise scaling factor for better performance.

4.1 Parameter Space Noise for Meta-RL

Injecting noise in high-dimensional neural network parameter space helps to escape local optima [Jin *et al.*, 2017]. Without loss of generality, we use Gaussian noise to perturb the parameters of deep neural networks for effective exploration. For our policy gradient RL case, parameter noise can be incorporated in gradient steps [Rückstieß *et al.*, 2008]. Given a policy $\pi_\theta(a|s)$ parameterized by $\theta \sim \mathcal{N}(\mu, \Sigma)$, with re-parametrization trick [Kingma and Welling, 2014], the gradient can be:

$$\nabla_{\mu,\Sigma}\mathbb{E}_\tau[R(\tau)] = \nabla_{\mu,\Sigma}\mathbb{E}_{\theta\sim\mathcal{N}(\mu,\Sigma)}[\sum_\tau p(\tau|\theta)R_t(\tau_i)] \quad (3)$$

$$\approx \frac{1}{N}\sum_{\epsilon_i,\tau_i}[\sum_{t=0}^{T-1}\nabla_{\mu,\Sigma}\log\pi(a_t|s_t;\mu+\epsilon_i\Sigma^{\frac{1}{2}})R_t(\tau_i)], \quad (4)$$

where we consider N samples $\epsilon_i \sim \mathcal{N}(0, I)$, and $\tau_i \sim (\pi_{\mu+\epsilon\Sigma^{\frac{1}{2}}}, p)$, where p is the transition probability. Re-parameterization trick decouples the mean, variance value and the stochasticity so that the original non-perturbed policy parameters can be updated with gradient steps. With this re-parameterization trick Plappert *et al.* [2018] show impressive exploration behavior on both high-dimensional discrete action tasks and continuous control tasks.

However, directly applying Gaussian noise and updating with re-parametrization trick as in Plappert *et al.* [2018] and Fortunato *et al.* [2018] requires lots of gradient steps to accurately estimate the unperturbed policy parameters. While in the meta-RL case, we focus on fast adaptation on new tasks and only a few steps are allowed. Instead of optimizing the original un-perturbed policy parameters during adaptation, we treat the perturbed as a variable to be optimized and directly fine-tune the perturbed models. The perturbed policy parameters are now $\tilde{\theta} = \theta + \epsilon_i\Sigma^{\frac{1}{2}}$ and are updated as follows:

$$\begin{aligned}\tilde{\theta} &= \tilde{\theta} + \beta\nabla_{\tilde{\theta}}\mathbb{E}_{\tau\sim\pi_{\tilde{\theta}}}[R(\tau)] \\ &= \tilde{\theta} + \beta\mathbb{E}_{\tau\sim\pi_{\tilde{\theta}}}[\sum_{t=0}^{T-1}\nabla_{\tilde{\theta}}\log\pi(a_t|s_t;\tilde{\theta})R_t(\tau_j)] \\ &= \tilde{\theta} + \beta\frac{1}{M}\sum_{j=1}^{M}[\sum_{t=0}^{T-1}\nabla_{\tilde{\theta}}\log\pi(a_t|s_t;\tilde{\theta})R_t(\tau_j)],\end{aligned} \quad (5)$$

where β is the gradient step size, and we collect M trajectories to estimate the policy gradient. Altogether we sample N samples $\epsilon_i \sim \mathcal{N}(0, I)$ to conduct N perturbed policies and fine-tune these policies according to Equation (5). Then, the best performing model is picked from the N perturbed networks.

4.2 Exploration with Structured Noise

For RL problems, compared with task-agnostic random noise, structured noise containing prior knowledge of the underlying task can help the learner better explore the insufficiently understood states. This has been shown in MAESN [Gupta *et al.*, 2018], which learns a task-specific latent variable z. $z \sim \mathcal{N}(\mu, \Sigma)$ containing the task-specific knowledge, is then concatenated to the states and used as structured prior noise for efficient and guided exploration. This also holds for parameter space noise. Hu *et al.* [2019] show that the parameters need to elaborate noise with prior knowledge rather than a task-agnostic random noise. In this part, we detail our Exploration with Structured Noise in Parameter Space (ESNPS) algorithm and apply the perturbed policy optimization strategy developed in Section 4.1.

In ESNPS, we partition meta-train tasks into J partitions $\mathcal{T} = \bigcup_{j=1}^{J} \mathcal{S}_j$ and conduct meta-training on each partition separately, resulting in altogether $J+1$ meta-policies $\{\pi_{\theta_j}\}_{j=1}^{J}$ (including one base model trained on the whole meta-train set). The intuition behind this is that the existing meta-learning algorithms would force the model to learn common structure (prior knowledge) from the training set [Finn *et al.*, 2018], but do not adequately model specific structures of different tasks [Lan *et al.*, 2019]. By partitioning the tasks and meta-train separately, meta-polices $\{\pi_{\theta_j}\}$ contain diverse prior knowledge. When used as structured noise, they potentially provide more task-specific information, and can be much more helpful for directed exploration on a new testing task than plain spherical noise. Specifically, we use the network meta-policy π_{θ_0} as a base model $\theta_0^{(1)}$, and other meta-polices $\{\pi_{\theta_j}\}$ as structured noise $\{\theta_j^{(1)}\}$. After adapted policy with Equation 1 on a specific task, we obtain fine-tuned models $\theta_0^{(2)}$, $\theta_j^{(2)}$ respectively. Then, we apply the following noise injection scheme:

$$\tilde{\theta}_j = \theta_0^{(2)} + \alpha\theta_j^{(2)}, \tag{6}$$

where $\tilde{\theta}_i$ is the perturbed model, and α is a scaling factor which will be discussed later. Then, as discussed in the previous part, we directly fine-tune the perturbed neural network $\tilde{\theta}_i$ with the help of structured noises for randomized but task-aware exploration:

$$\tilde{\theta}_j \leftarrow \tilde{\theta}_j + \beta\nabla_{\tilde{\theta}_j}\mathbb{E}[R(\tau_i)], \tag{7}$$

where β is the gradient step size. Besides the bonus of directed exploration based on the information contained in the structured noise, perturbing by parameter noise equals to changing the parameter position of policy parameter in the high-dimensional parameter space, and such a perturbation may help to escape the local minimum or saddle point in complex none-convex optimization problems [Jin *et al.*, 2017].

Like ordinary meta-learning algorithms, our ESNPS algorithm has two phases: meta-training and meta-testing. The architecture of ESNPS is depicted in Figure 1, and the algorithm is summarized in Algorithm 1. In the meta-training phase, we train $J+1$ MAML models on $J+1$ partitions $\{\mathcal{S}_i\}$ of the task set $\mathcal{T}$. In the meta-testing phase, we first fine-tune all the models with k gradient steps and put them into the Fine-tuned model collection. The number of gradient steps k for fine-tuning is always set to 4. We set model θ_0 trained on the whole training set $\mathcal{S}_0 = \mathcal{T}$ as the base model, and other J models as structured noises. Then, we iteratively perturb the base model θ_0 by using structured noises to generate diverse perturbed models. The parameters of perturbed models are computed by Equation 6. After further fine-tuning, we take the best performing perturbed model as a return. To the best of our knowledge, this is the first study to use the parameters of a learned neural network as structured noise.

Algorithm 1 ESNPS algorithm

Require: $J+1$ MAML models $\{\theta_0^{(1)}, \theta_1^{(1)}, \theta_2^{(1)}, \theta_3^{(1)}, \cdots, \theta_J^{(1)}\}$ trained from different partitions $\mathcal{S}_i$ of task set $\mathcal{T}$, where $\mathcal{T} = \bigcup_{j=1}^{J} \mathcal{S}_i$ and θ_0 trained on the whole task set $\mathcal{S}_0\mathcal{T}$.

```
Initialize fine-tuned model collection C = ∅
Initialize perturbed model collection E = ∅
for all MAML model θ ∈ {θ_0^(1), θ_1^(1), θ_2^(1), θ_3^(1), ..., θ_J^(1)} do
    Fine-tune θ with k gradient steps and put it to fine-tuned model collection C
end for
Set θ_0^(2) as the base model
for j = 1 to J do
    Perturb the base model θ_0^(2) by structured noise θ_i^(2) to generate perturbed model, put the perturbed model θ̃_i into perturbed model collection E.
end for
for all model θ ∈ E do
    Fine-tune θ with k gradient steps, put it into perturbed model collection E
end for
return The best performing perturbed model θ̃ from the perturbed model collection E
```

4.3 Noise Scaling

The noise level plays an important role in RL exploration. High-level noise encourages trials on insufficiently understood areas but potentially incurs high variance in the agent training. Noise level has been previously discussed in Plappert *et al.* [2018], where task-agnostic Gaussian noise is considered. In this section, we introduce a heuristic scheme that adaptively tunes the scaling factor α in our directed exploration ESNPS. Compared with perturbation in action space, injecting noise in parameter space is harder to depict. Let π and $\tilde{\pi}_j$ denote the original policy and the policy perturbed by noise model j. Following Plappert *et al.* [2018], we measure the distance between the two policies by the variance in action space they induce:

$$d(\pi, \tilde{\pi}_j) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}[\pi(s)_i - \tilde{\pi_j(s)}_i]^2} \tag{8}$$

where we average on M trajectories to estimate the distance between the two policies. As there are altogether J noise

models, the averaged distance $d(\pi, \tilde{\pi}) = \frac{1}{J}\sum_J d(\pi, \tilde{\pi}_j)$ is used to measure the averaged noise level. The scaling factor α then adaptively increases or decreases depending on whether the distance is below or above a certain threshold:

$$\alpha = \begin{cases} \lambda\alpha, & \text{if} \quad d(\pi, \tilde{\pi}) < \delta \\ \frac{1}{\lambda}\alpha, & \text{otherwise} \end{cases}, \tag{9}$$

where $\lambda \in \mathbb{R}^+$ is used to rescale α, which is set to 1.1 in our experiments. δ is a threshold controlling the acceptable change of actions due to noise injection.

With Equation (9), we change a policy parameter space searching problem into an action space searching problem, which can be much more easier to solve. In meta-RL, to get a certain level of exploration or perturbation, the proper value of scaling factor α can change rapidly for different tasks. However, as the meta-test tasks are often quite similar to each other, a certain level of exploration naturally requires a similar level of change in actions for the tasks, which means a shared threshold δ is enough for all the tasks.

This scaling factor searching scheme resembles the noise level adaptation strategy proposed in Plappert *et al.* [2018] in the formulation, but is for a different purpose and is carried out differently. We apply Equation (9) at the beginning of the policy gradient update of a specific task to get a proper α for further guided exploration. The scaling factor is kept unchanged during the latter exploration. In contrast, Plappert *et al.* [2018] focuses on a single task and dynamically varies the noise level during the whole exploration process.

5 Experiments

We evaluate the proposed ESNPS on four reinforcement learning tasks with MuJoCo simulator [Todorov *et al.*, 2012]. To show the effectiveness of guided search with structured noise, we compare ESNPS with vanilla MAML and a competitive parameter space task-agnostic noise model proposed in Plappert *et al.* [2018]. We omit the comparison with MAESN, which learns a task-specific latent variable for directed task-specific search because MAESN does not yield better performance in dense reward task.

5.1 Task Setup

Meta-RL tasks. To evaluate the proposed ESNPS algorithm, we experiment on four tasks: cheetah velocity, cheetah direction, ant velocity and ant direction. For all experiments, we use a neural network policy with two hidden layers of size 100, and ReLU nonlinearities. For the velocity tasks, the reward is the negative absolute value between the velocity given by the agent and the goal velocity. For direction tasks, the reward is the difference between the current direction and the goal direction. The goals of cheetah and ant are sampled uniformly from 0.0 to 2.0 and from 0.0 to 3.0, respectively. The horizon is set to $H = 200$, with 20 rollouts per gradient step for all groups of tasks except the ant direction task. The ant direction uses 40 rollouts. For each group of tasks, we construct 40 new tasks as meta-test set. For all the experiments, we exactly follow the protocol proposed in Finn *et al.* [2017].

Hierarchical partition. To acquire structured noise containing diverse prior knowledge, we hierarchically partition the meta-train tasks into J splits. $\mathcal{T} = \bigcup_{j=1}^{J} \mathcal{S}_j$. As meta-learning models exact common structures shared across meta-train tasks, these partitions, used as meta-train sets, can provide different levels of abstraction and diverse prior knowledge. Specifically, $\mathcal{S}_0 = \mathcal{T}$, and we bi-partition all the meta-train tasks into $\mathcal{S}_1$ and $\mathcal{S}_2$ according to the goal value of the tasks. Then, we equally partition the meta-train set into 4 splits forming $\{\mathcal{S}\}_{j=3}^{6}$ and finally 8 splits forming $\{\mathcal{S}\}_{j=7}^{14}$. For instance, $\mathcal{S}_1$ of cheetah velocity contains tasks with goals from $\mathcal{U}[0.0, 1.0)$ and $\mathcal{S}_2$ with goals from $\mathcal{U}[1.0, 2.0]$, where $\mathcal{U}$ is the uniform distribution. Similarly, tasks with goals from $\mathcal{U}[0.0, 0.5), \mathcal{U}[0.5, 1.0), \mathcal{U}[1.0, 1.5), \mathcal{U}[1.5, 2.0]$ form $\{\mathcal{S}\}_{j=3}^{6}$ respectively. Conducting meta-learning on $\{\mathcal{S}\}_{j=0}^{J}$ will result in meta-policies $\{\pi_{\theta_j}\}_{j=0}^{J}$. We set model π_{θ_0} as baseline MAML and perturb on policy parameters. In all our experiments, the fast adaptation Equation (1) is computed using vanilla policy gradient, and the meta-training is carried out with Trust Region Policy Optimization (TRPO) [Schulman *et al.*, 2015].

5.2 Performance Comparison

We get altogether 15 models after the hierarchical partition and meta-training, including a base model θ_0 trained on the whole meta-train set. Parameters of the other 14 meta-policies are used as structured noises to perturb the base model. In this section, we compare ESNPS with vanilla MAML and a parameter space Gaussian noise method [Plappert *et al.*, 2018] (denoted by PSNE). For a fair comparison, in the Gaussian noise case, we follow a similar noise injection and fine-tuning protocol: 14 random noises $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are used to perturb the base model θ_0, generating 14 perturbed models and the one with the best reward after fine-tuning is picked. Both structured noises (ESNPS) and Gaussian noises (PSNE) perturb the neural network in the parameter space. The perturbed policies are then updated as in Section 4.1. To get better performance, we adapt the scaling factor α of ESNPS with Equation (9) and the shared threshold δ controlling the change in action is set to be 1. For the Gaussian noise baseline PSNE, we set the noise level to $\sigma = 0.01$. This is a manually tuned noise level for PSNE. The properness of this value will be shown in the ablation study discussing the influence of noise level on final performance.

From Figure 2, we see that both ESNPS and PSNE improve the performance on vanilla MAML, validating our hypothesis that fine-tuning with parameter noise in high-dimensional neural network parameter space helps to escape local optima thus performing better exploration. While exploring with task-agnostic Gaussian noises only improves marginally, perturbing with the structured noise achieves significant improvement in all the four groups of tasks. Besides, we see that during the final fine-tuning, injecting Gaussian noise introduces large fluctuations while structured noise leads to much more stable training. We own this to the informative directed task-aware exploration in ESNPS.

In this experiment, every fine-tuning operation needs $k = 4$ gradient steps. Therefore, both the ESNPS and PSNE need

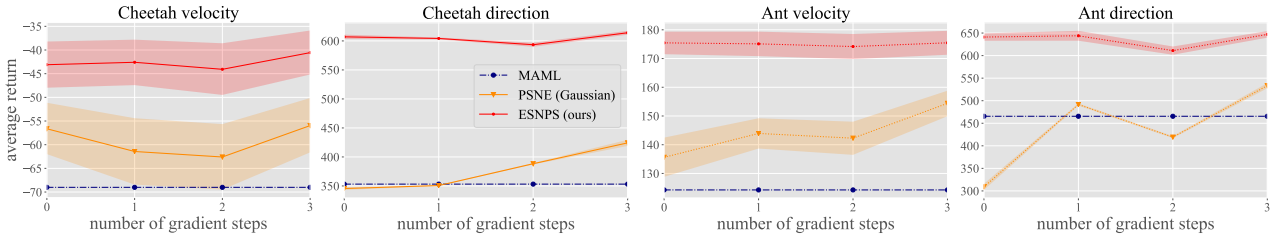

Figure 2: Comparison of ESNPS with baseline methods vanilla MAML and PSNE. For ESNPS, altogether 15 models are considered. Model θ_0 (MAML) is used as the base model and the other 14 models provide structured parameter space noises. PSNE also uses θ_0 as the base model, but perturbs it with plain spherical Gaussian noise instead of the structured noise containing previous learning experiences. The scaling factor α is obtained with a heuristic noise level adaptation scheme. The best reward is reported among the models perturbed by 14 different noises.

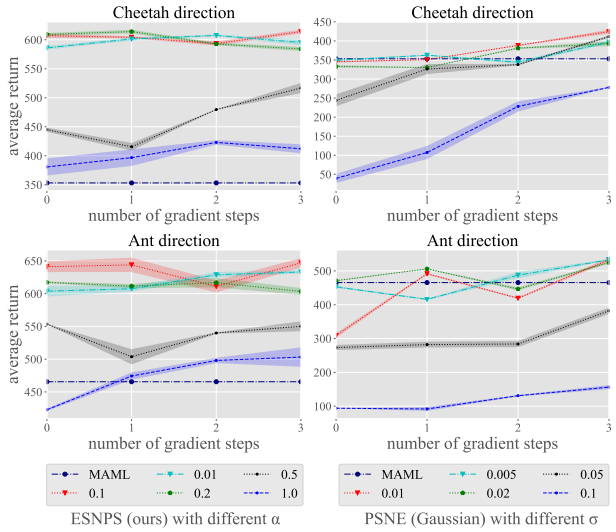

Figure 3: Effect of noise level. Left: ESNPS on cheetah and ant direction tasks with different scaling factor α. Right: PSNE on cheetah and ant direction tasks with different variance σ.

$15 * 4 + 14 * 4 = 116$ gradient steps, including the 4 steps of pre-adaptation for $\{\theta_j^{(2)}\}_{j=0}^{J}$ and the 4 steps of final adaptation of the perturbed model from $\{\tilde{\theta}_j\}_{j=1}^{J}$. From Figure 2 we see that the final fine-tuning steps on the perturbed $\tilde{\theta}_j$ do not always yield further performance gain, so that these fine-tuning steps are not always necessary. In this way, we can simplify ESNPS by running only the $15 * 4 = 60$ pre-adaptation gradient steps. From Figure 2 we see that even the simplified ESNPS (corresponding to the step 0 reward) always significantly outperforms PSNE, no matter whether the final fine-tuning steps are carried out or not. Compared with the baseline vanilla MAML, ESNPS obtains a significant improvement. With this notable achievement, we believe it is worthwhile even at the cost of more gradient steps.

5.3 Effect of Noise Level

Finally, we discuss how different noise levels will affect the exploration performance with the tasks of cheetah direction and ant direction. The other two groups of tasks are much simpler and can easily achieve high performance. We test the performance of ESNPS with different scaling factors α ranging from 0.01 to 1.0 (for all tasks, α is shared). Similarly, PSNE is evaluated with different variances σ ranging from 0.005 to 0.1. The results are shown in Figure 3. From Figure 3 we see that both ESNPS and PSNE are sensitive to the noise level and thus careful tuning of scaling factor α or variance σ is critical. In most cases, perturbing with structured noise shows far better exploration behavior than with Gaussian noise.

One interesting phenomenon is that when relatively high-level noise (e.g., $\alpha = 0.1$) is applied, ESNPS still performs comparably with vanilla MAML, indicating that the perturbed model still maintains the common structure of meta-train tasks. On the contrary, high-level noise introduces destructive perturbation when task-agnostic noise is applied, which validates the importance of task-specific guidance in exploration. Finally, note that with noise level $\sigma = 0.01$, the learner reaches the best performance in cheetah direction and explores comparably well in ant direction among the testing values, validating our setting of $\sigma = 0.01$ as the default variance level in Gaussian noise models in Section 4.2.

6 Conclusion and Future Work

In this paper, we introduce ESNPS to utilize previous search experiences as structured noise for directed exploration in new tasks. Experiments show the effectiveness of using diverse meta-policies as structured noise for exploration. Besides, we also demonstrate that directly fine-tuning the perturbed model is a suitable method for fast adaptive meta-learning methods like MAML. Although our method is widely applicable, it has limitations. For instance, constructing diverse structured noise is time-consuming. For future work, we plan to construct more easy-to-get yet powerful structured noises.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61832001 and No. 61672133) and Sichuan Science and Technology Program (No. 2019YFG0535 and No. 2018GZDZX0032).

References

[Bellemare *et al.*, 2016] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *NIPS*, pages 1471–1479, 2016.

[Brafman and Tennenholtz, 2001] Ronen I. Brafman and Moshe Tennenholtz. R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. In *IJCAI*, pages 953–958, 2001.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

[Finn *et al.*, 2018] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *NeurIPS*, pages 9537–9548, 2018.

[Fortunato *et al.*, 2018] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *ICLR*, 2018.

[Grant *et al.*, 2018] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas L. Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *ICLR*, 2018.

[Gupta *et al.*, 2018] Abhishek Gupta, Russell Mendonca, Yuxuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *NeurIPS*, pages 5307–5316, 2018.

[Hochreiter *et al.*, 2001] Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to learn using gradient descent. In *ICANN*, pages 87–94, 2001.

[Houthooft *et al.*, 2016] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: variational information maximizing exploration. In *NIPS*, pages 1109–1117, 2016.

[Hu *et al.*, 2019] Wenqing Hu, Chris Junchi Li, Lei Li, and Jian-Guo Liu. Near-optimal reinforcement learning in polynomial time. *Annals of Mathematical Sciences and Applications*, 4(1):3–32, 2019.

[Jin *et al.*, 2017] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *ICML*, pages 1724–1732, 2017.

[Kearns and Singh, 2002] Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Lan *et al.*, 2019] Lin Lan, Zhenguo Li, Xiaohong Guan, and Pinghui Wang. Meta reinforcement learning with task embedding and shared policy. In *IJCAI*, pages 2794–2800, 2019.

[Lemke *et al.*, 2015] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artif. Intell. Rev.*, 44(1):117–130, 2015.

[Li *et al.*, 2017] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *arXiv preprint*, arXiv:1707.09835, 2017.

[Mishra *et al.*, 2017] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-learning with temporal convolutions. *arXiv preprint*, arXiv:1707.03141, 2017.

[Mitchell and Thrun, 1992] Tom M. Mitchell and Sebastian Thrun. Explanation-based neural network learning for robot control. In *NIPS*, pages 287–294, 1992.

[Nichol *et al.*, 2018] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint*, arXiv:1803.02999, 2018.

[Osband *et al.*, 2016] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *NIPS*, pages 4026–4034, 2016.

[Pathak *et al.*, 2017] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *CVPR Workshops*, pages 488–489, 2017.

[Plappert *et al.*, 2018] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *ICLR*, 2018.

[Rückstieß *et al.*, 2008] Thomas Rückstieß, Martin Felder, and Jürgen Schmidhuber. State-dependent exploration for policy gradient methods. In *ECML/PKDD*, pages 234–249, 2008.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, pages 1889–1897, 2015.

[Sun *et al.*, 2011] Yi Sun, Faustino J. Gomez, and Jürgen Schmidhuber. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *AGI*, pages 41–51, 2011.

[Sutton and Barto, 2018] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2018.

[Tang *et al.*, 2017] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In *NIPS*, pages 2753–2762, 2017.

[Todorov *et al.*, 2012] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pages 5026–5033, 2012.

[Vilalta and Drissi, 2002] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artif. Intell. Rev.*, 18(2):77–95, 2002.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

[Yoon *et al.*, 2018] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *NeurIPS*, pages 7343–7353, 2018.