Adaptive Task Sampling and Variance Reduction for Gradient-Based Meta-Learning

Zhuoqun Liu
lzqyoooh@sjtu.edu.cn
Yuankun Jiang
yuankunjiang@sjtu.edu.cn
Chenglin Li
lcl1985@sjtu.edu.cn
Wenrui Dai
daiwenrui@sjtu.edu.cn
Junni Zou
zoujunni@sjtu.edu.cn
Hongkai Xiong
xionghongkai@sjtu.edu.cn

Shanghai Jiao Tong University
800 Dongchuan Road,
Shanghai 200240, China

Abstract

Meta-learning enables fast adaptation of the trained model to new tasks by exploiting the similarity between tasks sampled for training, leading to its success in few-shot learning and domain adaptation. Conventional meta-learning paradigm, however, treats different tasks equally important and thus samples them at a uniform distribution for training, which may result in a sub-optimal performance with high variance introduced to the gradients. To address this, in this paper, we develop a novel adaptive task sampling and variance reduction (ATSVR) method for gradient-based meta-learning. Built upon gradient-based meta-learning framework, we are able to assign different importance weights to the training tasks, by leveraging the importance sampling technique to approximately manipulate the sampling distribution according to a target distribution that is updated iteratively towards minimising the meta-objective. In addition, update of this target distribution is also enforced to reduce the variance of gradient estimate at each iteration. Empirical evaluations on a regression task demonstrate the performance gain by introducing adaptive task sampling to meta-learning, while those on the few-shot learning task on two benchmarks show that our ATSVR outperforms state-of-the-art adaptive sampling-based baselines, such as meta-learning with adaptive task scheduler.

1 Introduction

In the field of artificial intelligence, there has long been a desire to facilitate deep learning models with the ability of rapid learning with a small number of samples, just as humans do. However, the learning processes of humans and current neural networks are still significantly different. While humans can quickly learn a new task with only a few attempts by leveraging the prior knowledge gained from similar tasks, it usually requires enormous data samples for training a neural network to well adapt to the new task. Meta-learning, also known as learning to learn [4], has been recently proposed as a common solution to narrow this gap. It leverages the relevant information obtained from training of previous tasks, and thus can achieve a more efficient learning on new tasks with a much faster adaptation.

In literature, there are two dominant strands of meta-learning methods: gradient-based [7] and metric-based [18]. Among them, the gradient-based meta-learning has received more attention due to its general applicability to both the regression and few-shot classification tasks. As a common baseline for gradient-based meta-learning, model-agnostic meta-learning (MAML) [7] attempts to learn a set of initialisation parameters that are expected to contain a priori knowledge of each task in the task space. In this way, if this set of initialisation parameters are applied to the model for learning a new task later in the meat-test stage, the experience stored therein can be leveraged to enable a fast adaptation. Specifically during training, MAML and other gradient-based meta-learning methods treat different tasks equally important and thus sample them at a uniform distribution over the task space. This corresponds to assigning an identical importance to all tasks in the entire period of training, which, however, has been shown to incur a sub-optimal performance [2, 23], since the contribution of different tasks to the model's training should be different and varying at different stages of the training. Due to the heterogeneity of tasks, on the other hand, the multiple local update steps implemented for a specific task may further result in a high variance in the update direction of meta learner [5]. This will drive the learned model to overfit to a subset of tasks sampled for training, losing the ability to generalise to the entire task space.

To address the above problems, in this paper, we develop a novel adaptive task sampling and variance reduction (ATSVR) scheme for gradient-based meta-learning. Compared to conventional gradient-based meta-learning, our scheme can adaptively adjust the sampling distribution of the task space w.r.t. the training process based on the meta-objective. As the meta-training proceeds, we assign different importance weights to different tasks, by leveraging the importance sampling (IS) [8] technique to approximately manipulate the sampling distribution according to a target distribution that is updated iteratively towards minimising the meta-objective. To additionally avoid overfitting to some partial tasks, update of this target distribution is further enforced to reduce the variance of gradient estimate at each iteration, thus improving the generalisation capability for the learned model. We finally evaluate our ATSVR on a toy regression task and the practical few-shot image classification task. The regression experiments demonstrate that ATSVR outperforms MAML, while the variance of gradients is further reduced greatly. In the few-shot learning tasks, compared to ATS [23] (state-of-the-art adaptive sampling-based baseline), our ATSVR improves the average classification accuracy by 0.68% and 0.31% under the 1-shot and 5-shot settings on miniImageNet with limited budget. On the multidataset benchmark, ATSVR outperforms ATS on 3 out of the 4 datasets, with average accuracy gains of 0.16% and 1.05% under the 1-shot and 5-shot settings, respectively.

Our main contributions can be summarised as follows.

- We propose an adaptive task sampling scheme to adaptively adjust the sampling distribution during meat-training, which assigns different importance weights to different tasks based on a target distribution updated towards minimising the meta-objective.

- We reduce the variance of gradient estimate when updating the target distribution, thus mitigating the overfitting issue and improving the generalisation capability.

- Empirical evaluations on both regression and few-shot learning tasks validate the benefit introduced by the proposed adaptive task sampling and variance reduction scheme.

2 Related Work

Gradient-Based Meta-Learning. As a typical strand of meta-learning methods, gradient-based meta-learning has recently attracted broad research interests because of its general applicability to different types of tasks. Finn *et al.* in [7] propose MAML, aiming at learning the optimal initialization of a base learner, which can quickly adapt its performance to new tasks with only a few gradient-based update steps. The original MAML requires computation of higher-order gradients in the back-propagation during the adaptation phase. To alleviate the computational burden, they further propose a variant of MAML, named FOMAML [7], where the computation of second-order gradients is abandoned at the cost of only a slight decline in the model's performance. Nichol *et al.* in [15] design the Reptile algorithm to leverage the average update direction in the adaptation phase as the update direction of meta-update, which is more robust than FOMAML. To further reduce the computational complexity, Raghu *et al.* in [17] consider updating only partial model parameters during the adaptation phase, while final performance of the base leaner is demonstrated to be not significantly affected. A common drawback of the above methods is that tasks are sampled uniformly in meta-training. Since tasks in the training set does not necessarily follow a uniform distribution over the entire task space, uniform sampling of them may drive the learned model to overfit to a subset of tasks used for training and lose the ability to generalise.

Task Sampling Method. Some recent works have focused on optimizing the sampling distribution of tasks for meta-learning. In meta-reinforcement learning, Jabri *et al.* in [9] propose to generate tasks based on the reward function. The authors in [10, 13] further consider reconstructing the task sampling distribution in terms of the amount of information, where the probability of each task being selected is proportional to the amount of information it provides. In supervised meta-learning, most existing methods borrow ideas from the curriculum learning [3], and adjust the sampling distribution w.r.t. task difficulty. For example, Li *et al.* in [11] propose a difficulty-aware loss function for meta-learning. Liu *et al.* in [12] use the idea of greed to construct class-pairs for building the training task set. Recently, Arnold *et al.* in [2] propose to perform a uniform sampling alternatively in the task difficulty space. They first demonstrate that the learning difficulty of different tasks follows a normal distribution for arbitrary datasets and network structures, and then improve the model's generalization capability by sampling uniformly over the task difficulty space based on importance sampling. Though these methods have adjusted the sampling distribution of tasks from different perspectives, their sampling strategies should all be set manually before training. In fact, due to the uncertainty of training, one may expect the sampling strategy to evolve with the training process such that it can well adapt to different stages of training. Following this idea, Yao *et al.* in [23] propose to adaptively adjust the sampling strategy by training an adaptive task scheduler (ATS) using reinforcement learning. ATS takes the current model's loss on the validation set as input, and assigns different weights to the tasks sampled at each round to change the sampling distribution. In summary, most of these existing methods attempt to adjust the sampling distribution based on task difficulty, while we aim to adjust it directly in the task space. This may avoid inconsistency in sampling distribution obtained in the task difficulty space, which is caused by changes of the task difficulty as training progresses.

3 Preliminaries

Gradient-Based Meta-Learning. Suppose that we are given a task set $\{\mathcal{T}_i\}$ with corresponding task distribution denoted by $p(\mathcal{T})$. During meta-training, the meta learner aims to extract common knowledge from the task set $\{\mathcal{T}_i\}$ and learn the optimal model parameter θ^* for a base learner, which can quickly improve its performance on new tasks within a few gradient update steps later in the meta-test. Our main focus in this paper is to consider the few-shot learning setting, where a dataset with limited number of samples $\{x_{i,j}, y_{i,j}\}_{j=1}^n$ can be accessed for each task $\mathcal{T}_i$. This dataset is further divided into $\mathcal{D}_{\mathcal{T}_i}^{tr}$ and $\mathcal{D}_{\mathcal{T}_i}^{val}$ for training and validation, respectively.

Our approach builds upon MAML, the gradient-based meta-learning baseline. The meta-training objective for supervised few-shot learning can thus be written as:

$$\min_{\theta} J(\theta) = \mathop{\mathbb{E}}_{\mathcal{T}\sim p(\mathcal{T})}\left[\mathcal{L}\left(\mathcal{D}_{\mathcal{T}}^{val}; \theta'_{\mathcal{T}}\right)\right] \quad \text{s.t. } \theta'_{\mathcal{T}} = \theta - \alpha \nabla_{\theta} \mathcal{L}\left(\mathcal{D}_{\mathcal{T}}^{tr}; \theta\right), \tag{1}$$

where $\mathcal{L}(\cdot)$ can be the MSE loss in regression tasks, or the cross entropy loss in few-shot classification tasks. MAML performs a nested loop to solve the optimization problem in Eq. (1). In the inner loop, the base learner updates its model parameter $\theta'_{\mathcal{T}}$ for task $\mathcal{T}$, by performing the gradient-based update on training dataset $\mathcal{D}_{\mathcal{T}}^{tr}$ w.r.t. the meta model parameter θ. This updated model $\theta'_{\mathcal{T}}$ is then used to compute the loss on validation dataset $\mathcal{D}_{\mathcal{T}_i}^{val}$. In the outer loop, MAML aggregates the loss from all the tasks by taking expectation over the entire task set w.r.t. the task sampling distribution. The meta model is therefore updated by:

$$\theta = \theta - \beta \nabla_{\theta} \mathop{\mathbb{E}}_{\mathcal{T}\sim p(\mathcal{T})}\left[\mathcal{L}\left(\mathcal{D}_{\mathcal{T}}^{val}; \theta'_{\mathcal{T}}\right)\right]. \tag{2}$$

To reduce computational complexity and speed up the meta-training, the almost no inner loop (ANIL) is proposed in [17] as a practical variant of MAML, which only updates the last layer while freezing the rest layers of neural network in the inner loop. Experiments show that ANIL significantly reduces the training time but with only a slight decrease in accuracy.

Task Representation Learning. In supervised learning, each task is characterized by a dataset, where the measurement of correlation (similarity) between tasks is intractable, especially in image classification. Task representation learning [22] aims to provide such a correlation measurement for tasks in the latent space. For a certain task $\mathcal{T}_i$, an embedding function $\mathcal{F}(x, y)$ first embeds each sample $(x_{i,j}, y_{i,j})$. The sample embeddings are then fed into a recurrent aggregator $\mathcal{A}(\mathcal{F}(\cdot))$ to obtain a highly effective representation $\mathbf{g}_{i,j}$ for each sample. The representation of task $\mathcal{T}_i$ is defined by taking the average of representations of all samples: $\tau_i = \frac{1}{n}\sum_j^n(\mathbf{g}_{i,j})$. In the rest of this paper, we use both τ and $\mathcal{T}$ to interchangeably denote a certain task with slight ambiguity.

4 Adaptive Task Sampling and Variance Reduction

In this section, we introduce the proposed adaptive task sampling and variance reduction (ATSVR) for gradient-based meta-learning, to tackle the sub-optimality stemmed from the uniform sampling over training task set during meta-training. We first present adaptive task sampling for meta-training objective based on importance sampling (IS). We then formulate the variance reduction problem for the gradient estimate of gradient-based meta-learning.

4.1 Importance Sampling-Based Adaptive Task Sampling

In adaptive task sampling, we aim to adjust the current sampling distribution $p(\mathcal{T})$ to a certain target distribution $q(\mathcal{T};\phi)$ that is parameterized by ϕ. The importance sampling estimator provides an unbiased estimate of the original meta-objective value under distribution $p(\mathcal{T})$, by weighing the loss of each task with an importance ratio. The importance sampling-based meta-training problem with a task sampling distribution shift can be formulated as:

$$\min_{\theta} J^{IS}(\theta) = \underset{\mathcal{T}\sim p(\mathcal{T})}{\mathbb{E}} \left[w(\mathcal{T};\phi)\mathcal{L}\left(\mathcal{D}^{val}_{\mathcal{T}};\theta_{\mathcal{T}}\right)\right] \quad \text{s.t. } \theta'_{\mathcal{T}} = \theta - \alpha\nabla_{\theta}\mathcal{L}\left(\mathcal{D}^{tr}_{\mathcal{T}};\theta\right), \tag{3}$$

where $w(\mathcal{T};\phi) \triangleq \frac{q(\mathcal{T},\phi)}{p(\mathcal{T})}$ denotes the importance ratio. It is not difficult to verify the unbiasedness of the estimated objective in Eq. (3) to the original objective in Eq. (1). With the IS-based optimization formulation in Eq. (3), we are able to approximately apply the target distribution for sampling without changing the actual sampling distribution. Given a task batch of size $\mathcal{B}$ at each iteration, we then determine the following empirical meta-objective:

$$\min_{\theta} \hat{J}^{IS}(\theta) = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\frac{q(\mathcal{T}_i,\phi)}{p(\mathcal{T}_i)}\mathcal{L}\left(\mathcal{D}^{val}_{\mathcal{T}_i};\theta_{\mathcal{T}_i}\right) = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} w\left(\mathcal{T}_i,\phi\right)\mathcal{L}\left(\mathcal{D}^{val}_{\mathcal{T}_i};\theta_{\mathcal{T}_i}\right). \tag{4}$$

Here, we aim to dynamically adjust the target distribution to consistently approach the optimum, which thus motivates our formulation of the following optimization problem for optimizing the target distribution $q(\cdot;\phi)$ together with the meta model θ:

$$\min_{\theta;\phi} \hat{J}^{ADP}(\theta,\phi) = \sum_{i=1}^{|\mathcal{B}|}\overline{w}\left(\mathcal{T}_i,\phi\right)\mathcal{L}\left(\mathcal{D}^{val}_{\mathcal{T}_i};\theta_{\mathcal{T}_i}\right), \tag{5}$$

where $\overline{w}\left(\mathcal{T}_i,\phi\right) \triangleq \frac{w(\mathcal{T}_i,\phi)}{\sum_{j=1}^{|\mathcal{B}|} w(\mathcal{T}_j,\phi)}$ is the weighted importance sampling (WIS) [16] ratio. As a variant of IS, WIS has two advantages during the optimization of Eq. (5). First, the utilization of $\overline{w}\left(\mathcal{T}_i,\phi\right)$ gets rid of the trivial full-zeros solution when simply using $w\left(\mathcal{T}_i,\phi\right)$. This is because when $w\left(\mathcal{T}_i,\phi\right) \to 0$ for all the tasks, we have $\overline{w}\left(\mathcal{T}_i,\phi\right) \to \frac{1}{|\mathcal{B}|}$ for each task and thus the sampling strategy degrades to a uniform distribution. Second, the WIS estimator has a lower variance than IS but with a sacrifice in the unbiasedness. Though WIS is biased, it is asymptotically correct since $\mathbb{E}[w(\mathcal{T},\phi)] = 1$ [19]. By updating the target distribution ϕ with gradient $\nabla_{\phi}\hat{J}^{ADP}$ at each iteration during meta-training, we are adaptively adjusting it towards the direction of minimising the meta-objective.

In the few-shot classification problem, there is no way of getting a representation for each task in the task space. This makes it even difficult for us to change the sampling distribution in the task space. We therefore consider using the aggregator in [22] to learn the representation of a task. At each iteration, we feed the sampled tasks to the recurrent autoencoder aggregator $\mathcal{A}(\mathcal{F}(\cdot))$, and obtain the corresponding task representation τ. Hence, in practice we utilize a function $f_{\phi}(\tau_i)$ to approximate $w\left(\mathcal{T}_i,\phi\right)$, which is learned by a shallow neural network with sigmoid as the activation function in the last layer. We can then formulate the following meta-objective for the few-shot classification problem:

$$\min_{\theta;\phi} \hat{J}^{FS}(\theta,\phi) = \sum_{i=1}^{|\mathcal{B}|}\bar{f}_{\phi}(\tau_i)\mathcal{L}\left(\mathcal{D}^{val}_{\mathcal{T}_i};\theta_{\mathcal{T}_i}\right), \quad \bar{f}_{\phi}(\tau_i) = \frac{f_{\phi}(\tau_i)}{\sum_{j=1}^{|\mathcal{B}|} f_{\phi}(\tau_j)}. \tag{6}$$

4.2 Variance Reduction for Gradient-Based Meta-Learning

Gradient-based method is known to suffer high variance [5]. In the gradient update of meta model, the estimate of gradient $\nabla_\theta J(\theta)$ is expected to have additional source of variability, which is incurred by the randomly sampled tasks. We are thus motivated here to further reduce the gradient variance to accelerate the convergence of meta model, by regularizing the update of ϕ towards the direction of decreasing the variance of gradient estimate.

To solve Eq. (5) in the outer loop, the gradient estimate of meta model can be written as:

$$\nabla_\theta \hat{J}^{ADP}(\theta,\phi) = \sum_{i=1}^{|\mathcal{B}|} \overline{w}(\mathcal{T}_i,\phi)\nabla_\theta \mathcal{L}\left(\mathcal{D}_{\mathcal{T}_i}^{val};\theta_{\mathcal{T}_i}\right), \tag{7}$$

which has variance w.r.t. task sampling distribution $p(\mathcal{T})$ satisfying the following inequality:

$$\mathrm{Var}_{p(\mathcal{T})}\left(\nabla_\theta \hat{J}^{ADP}(\theta,\phi)\right) \leq |\mathcal{B}|^2 v(\phi), \quad v(\phi) = \mathrm{Var}_{p(\mathcal{T})}\left(\overline{w}(\mathcal{T},\phi)\nabla_\theta \mathcal{L}\left(\mathcal{D}_{\mathcal{T}}^{val};\theta_{\mathcal{T}}\right)\right). \tag{8}$$

Proof 1 *The expectation of $\nabla_\theta \hat{J}^{ADP}(\theta,\phi)$ w.r.t. $p(\mathcal{T})$ is given by:*

$$\mathbb{E}_{p(\mathcal{T})}[\nabla_\theta \hat{J}^{ADP}(\theta,\phi)] = \sum_{i=1}^{|\mathcal{B}|} \mathbb{E}_{p(\mathcal{T})}\left[\overline{w}(\mathcal{T}_i,\phi)\nabla_\theta \mathcal{L}\left(\mathcal{D}_{\mathcal{T}_i}^{val};\theta_{\mathcal{T}_i}\right)\right]. \tag{9}$$

Hence, we can expand the variance as follows:

$$\begin{aligned}
&\mathrm{Var}_{p(\mathcal{T})}\left(\nabla_\theta \hat{J}^{ADP}(\theta,\phi)\right) \qquad (10)\\
=&|\mathcal{B}|^2\mathbb{E}\left[\left\|\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\overline{w}(\mathcal{T}_i,\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}_i}^{val};\theta_{\mathcal{T}_i}) - \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\mathbb{E}_{p(\mathcal{T})}[\overline{w}(\mathcal{T},\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}}^{val};\theta_{\mathcal{T}})]\right\|\right]^2\\
=&|\mathcal{B}|^2\mathbb{E}\left[\left\|\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left(\overline{w}(\mathcal{T}_i,\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}_i}^{val};\theta_{\mathcal{T}_i}) - \mathbb{E}_{p(\mathcal{T})}[\overline{w}(\mathcal{T},\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}}^{val};\theta_{\mathcal{T}})]\right)\right\|\right]^2\\
\leq&|\mathcal{B}|\sum_{i=1}^{|\mathcal{B}|}\mathbb{E}\left(\|\overline{w}(\mathcal{T}_i,\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}_i}^{val};\theta_{\mathcal{T}_i}) - \mathbb{E}_{p(\mathcal{T})}[\overline{w}(\mathcal{T},\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}}^{val};\theta_{\mathcal{T}})]\|\right)^2\\
=&|\mathcal{B}|\sum_{i=1}^{|\mathcal{B}|}Var_{p(\mathcal{T})}\left(\overline{w}(\mathcal{T},\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}}^{val};\theta_{\mathcal{T}})\right)\\
=&|\mathcal{B}|^2 Var_{p(\mathcal{T})}\left(\overline{w}(\mathcal{T},\phi)\nabla_\theta\mathcal{L}(\mathcal{D}_{\mathcal{T}}^{val};\theta_{\mathcal{T}})\right),
\end{aligned}$$

where the inequality in Eq. (10) follows Jensen's inequality.

To reduce the variance of gradient estimate in the update of meta model, based on our theoretical findings in Eq. (8), we propose to solve the following optimization problem:

$$\min_\phi \hat{J}^{ADP}(\theta,\phi) + \lambda v(\phi), \tag{11}$$

where we jointly consider the performance improvement of supervised learning (i.e., first term in Eq. (11)) and reduction of the gradient estimate's variance (i.e., second term in Eq.

Algorithm 1: Adaptive task sampling and variance reduction (ATSVR) algorithm.

Input: distribution over tasks: $p(\mathcal{T})$, task weight function parameterized by ϕ: $f_\phi(\mathcal{T})$, learning rate: α_1 and α_2, weight of the variance term: λ

Randomly initialize θ, ϕ;

while *not done* **do**

  Sample a batch of tasks from $p(\mathcal{T})$ uniformly;

  for *all tasks* **do**

    Obtain the representation τ of the task through a pretrained recurrent aggregator $\mathcal{A}(\mathcal{F}(.))$, which is fed into $f_\phi(\tau)$ to obtain task weight;

    Compute adapted parameters with gradient descent:

    $\theta'_{\mathcal{T}} = \theta - \alpha_1 \nabla_\theta \mathcal{L}(\mathcal{D}^{tr}_{\mathcal{T}}; \theta)$;

  end

  Update $\theta \leftarrow \theta - \alpha_2 \nabla_\theta \hat{J}^{ADP}(\theta, \phi)$;

  Update $\phi \leftarrow \phi - \alpha_2 \nabla_\phi \hat{J}^{ADP}(\theta, \phi) - \lambda \nabla_\phi Var(\phi)$;

end

(11)). Based on our theory, we propose the adaptive task sampling and variance reduction (ATSVR) scheme for gradient-based meta-learning, which is summarised in Algorithm 1. Our ATSVR algorithm is built upon MAML, with the yellow shaded area showing our unique contribution in this paper. In ATSVR, we sample the task uniformly, as in MAML. In the few-shot classification task, we first feed the tasks into a pretrained aggregator $\mathcal{A}(\mathcal{F}(\cdot))$ to get a representation τ for each task, and feed that representation into $f_\phi(\tau)$ to get the weights belonging to each task. In the inner loop update, we compute the task specific-parameter in the same way as in MAML. While in the outer loop update, the normalised weights are assigned to different tasks to form a new meta-objective for updating θ and ϕ. In addition, the variance of gradient estimate is used as an additional regularization term for updating ϕ.

5 Experiment

5.1 Regression

Settings. In this experiment, we consider a sinusoidal curve $y(x) = A\sin(\omega x + b)$ as the target function, where the amplitude A, frequency ω, and phase b follow the uniform distribution over intervals $[0.1, 5.0]$, $[0.8, 1.2]$, and $[0, \pi]$, respectively. The input range is restricted within the interval $[-5.0, 5.0]$. For the regressor and hyperparameters such as learning rate and batch size, we use the same settings as in [7].

In the regression task, the representation of each task in the task space can be obtained directly from the magnitude A, phase b and frequency ω of the sinusoidal curve. Thus, we use $[A_i, b_i, \omega_i]$ directly as the representation for task $\mathcal{T}_i$. To better illustrate our ATSVR, instead of using neural networks, we adopt a multi-dimensional Gaussian distribution as the target distribution $q(\mathcal{T}; \phi)$, where ϕ denotes the mean and variance of the Gaussian distribution. The original distribution $p(\mathcal{T})$ is a three-dimensional uniform distribution in the amplitude, phase and frequency dimension. To prevent too much deviation between the original and target distributions, which may lead to a high variance of important sampling, we use the KL

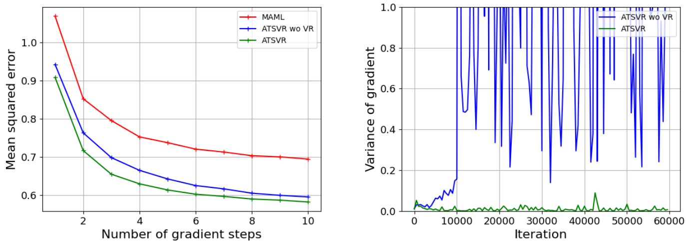

Figure 1: **Left:** Mean squared error (MSE) loss at different steps of the inner-loop updates for 5-shot regression; **right:** variance of gradients during training.

		MAML [7]	ATSVR (w/o VR)	ATSVR
10-shot	pre-update	3.058 ±0.23	3.135 ±0.23	3.11 ±0.24
	post-update	0.73 ±0.06	0.61 ±0.06	**0.56 ± 0.06**
5-shot	pre-update	3.159 ±0.28	3.175 ±0.28	3.205 ±0.29
	post-update	1.06 ±0.12	0.93 ±0.10	**0.90 ± 0.10**

Table 1: MSE of sinusoidal curve regression, where pre-update and post-update refer to the MSE before and after one step gradient update on the test dataset, respectively.

divergence between the original and target distributions as an additional objective function:

$$\phi \leftarrow \phi - \alpha_2 \nabla_\phi \hat{J}^{ADP}(\theta, \phi) - \lambda \nabla_\phi Var(\phi) - \gamma \nabla_\phi KL(p(\mathcal{T}_i)||q(\mathcal{T}_i, \phi)). \tag{12}$$

Here we set both λ and γ to 1. In addition, our ATSVR algorithm shows that when calculating the variance of gradients, we need to calculate the outer-loop gradient for each task in the batch in the outer loop. This is unacceptable in terms of computational complexity for second-order gradient-based algorithms, such as MAML and ANIL. To reduce the computational complexity, we thus refer to FOMAML and use the first-order gradient to approximate the second-order gradient. Specifically, when computing the gradient of each task in the outer loop, we follow FOMAML and discard the second-order gradient, using the last update direction of the task in the inner loop as the gradient of the outer loop. In this way, the algorithm's complexity is greatly reduced.

We adopt MAML as the meta-learner backbone, and compare our ATSVR algorithm with MAML in $K \in [5, 10]$ shot setting. Here, both MAML and our ATSVR use one-step adaptation, and are trained for 60,000 iterations with a meta batch-size of 4 tasks.

Results. For performance evaluation, we randomly sample 600 sinusoidal curves. For each curve, we sample K examples for training and another K examples for testing. We repeat this procedure 600 times and take the average of mean squared error (MSE). The results with 95 confidence intervals are summarized in Table 1. It can be seen that our ATSVR achieves a lower MSE on the post-update than MAML, in both the 5-shot and 10-shot cases. In addition, we also validate the effectiveness of our proposed adaptive task sampling and variance reduction separately. ATSVR (w/o VR) denotes our ATSVR by removing the variance reduction term from the objective function when updating parameter ϕ. Experiments show that by removing the variance reduction term, ATSVR still has an advantage over MAML. Additionally, the left of Fig. 1 shows how the MSE loss changes with the number of inner-loop updates. It is worth noting that we start counting the MSE after the first update, which

Model	5-way 1-shot	5-way 5-shot
ANIL [17]	33.61 ±0.66%	45.97 ±0.65%
GCP [12]	34.69 ±0.67%	46.86 ±0.68%
PAML [10]	33.64 ±0.62%	45.01 ±0.69%
DAML [11]	34.83 ±0.69%	46.66 ±0.67%
USOD [2]	34.70 ±0.50%	46.70 ±0.70%
ATS [23]	35.15 ±0.67%	47.76 ±0.68%
ATSVR	**35.83 ± 0.72%**	**48.07 ± 0.65%**

Table 2: Performance on miniImageNet with limited budget 16.

Model / Budget	16	32	48	64
ANIL [17]	33.61 ±0.66%	40.48 ±0.75%	44.07 ±0.80%	45.73 ±0.79
ATS [23]	35.15 ±0.67%	41.68 ±0.78%	44.89 ±0.79%	46.27 ±0.80%
ATSVR	**35.83 ± 0.72%**	**43.83 ± 0.69%**	**45.80 ± 0.82%**	**47.76 ± 0.77%**

Table 3: Performance on miniImageNet with different limited budgets.

has already undergone a round of gradient update and corresponds to the post-update value in Table 1. It can be seen that ATSVR has a better convergence compared to MAML. The right of Fig. 1 further shows how variance of gradients changes during training. It can be seen that during ATSVR's training, the variance of gradients is effectively reduced.

5.2 Few-Shot Image Classification

Settings. In the few-shot classification task, we validate the effectiveness of our ATSVR on two benchmark datasets: miniImageNet with limited budget [23] and multidataset [22]. Here, miniImageNet with limited budget indicates that the number of classes is limited. The original miniImageNet has 64 trainig image classes. Following the settings in [23], in this experiment we reduce the number of classes to 16, resulting in 4,368 5-way combinations. For multidataset, we follow the settings in [22], which consists of four image classification datasets: Caltech-UCSD Birds-200-2011 (Bird) [21], Describable Textures Dataset [6], Fine-Grained Visual Classification of Aircraft [14], and FGVCx-Fungi (Fungi) [1]. The images are pre-processed in the same way as in [22]. For both datasets, we pre-train a recurrent autoencoder aggregator $\mathcal{A}(\mathcal{F}(\cdot))$ to learn the representation of tasks, with the dimensionality of representation vector set to 128. The target distribution $q(\mathcal{T};\phi)$ is learned by a neural network with 2 hidden layers of size 40 and sigmoid as the activation of the last layer. The weight of variance reduction term λ is set to 2. For the calculation of gradient of each task to get the variance, we use the same method as in the regression experiment. We compare ATSVR with existing works in the field of meta-learning that perform sampling distribution optimization, including GCP [12], DAML [11], ATS [23], USOD [2], and PAML [10]. To be consistent with the settings in these methods, we use ANIL [17] as the meta learner and set the network to a standard four-block convolutional architecture with 32 filters [20].

Results. For performance evaluation, we randomly sample 1000 tasks as the test tasks. Results on the miniImageNet dataset with limited budget 16 are summarized in Table 2. It can be seen that our ATSVR outperforms existing methods that perform sampling distribution optimization for meta-learning in both the 1-shot and 5-shot settings. Further, we study the impact of different limited bugets on the test results in the 1-shot setting, with results summarized in Table 3. It can be seen that our ATSVR outperforms the currently best adaptive

5-way 5-shot					
Model	Bird	Texture	Aircraft	Fungi	Average
ANIL [17]	69.52 ±0.78%	41.54 ±0.57%	59.27 ±0.67%	51.31 ±0.83%	55.41%
ATS [23]	66.95 ±0.78%	**42.24 ± 0.62%**	62.14 ±0.71%	51.05±0.79%	55.59%
ATSVR	**70.59 ± 0.77%**	41.95 ±0.73%	**62.49 ± 0.67%**	**51.52 ± 0.83%**	**56.64%**
5-way 1-shot					
Model	Bird	Texture	Aircraft	Fungi	Average
ANIL [17]	51.20 ±0.97%	31.20 ±0.65%	46.28 ±0.82%	38.75 ±0.86%	41.86%
ATS [23]	51.34 ±0.95%	30.69±0.63%	**47.67 ± 0.80%**	39.22±0.86%	42.23%
ATSVR	**52.03 ± 0.97%**	**31.33 ± 0.66%**	46.72 ±0.86%	**39.47 ± 0.89%**	**42.39%**

Table 4: Performance on the multidataset benchmark.

sampling method ATS [23], under different budget settings. This demonstrates the better generalisation capability of our approach to the test dataset. In Table 4, we further compare ATSVR with ANIL and ATS on multidataset benchmark. It can be seen that ATSVR outperforms ATS in three out of the four datasets, in both the 1-shot and 5-shot settings. It is also worth mentioning that, we conduct a significance test on our experimental results based on one tailed t-tests, which shows that our ATSVR has a 70% confidence level better than ATS on most problems.

6 Conclusion and Future Work

We proposed the adaptive task sampling and variance reduction (ATSVR) scheme for gradient-based meta-learning, which could adaptively adjust the sampling distribution of the task space with the training process based on the meta-objective. To prevent the model from eventually overfitting to some tasks due to the heterogeneity of tasks and incomplete sampling at each iteration, we further reduced the variance of gradient estimate at each iteration by updating the target distribution. Empirical evaluations on both regression and few-shot learning tasks have validated the effectiveness of our proposed ATSVR scheme.

We found that it might be a promising future work to combine our method with meta-RL, as follows. **1)** In meta-RL, variance of the gradient estimations is usually higher when the policy gradient is updated because of the different settings of environment's dynamic rewards. While our ATSVR can calculate the upper bound of the variance of gradient estimation for meta-RL. Then, by dynamically adjusting the weights of different environment settings, this upper bound is continuously reduced during the training process. **2)** Many existing meta-RL methods were implemented based on MAML framework, and our ATSVR method can thus be extended to these methods easily.

7 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61871267, Grant 61831018, Grant 61972256, Grant T2122024, Grant 61971285 and Grant 62120106007, in part by the Program of Shanghai Science and Technology Innovation Project under Grant 20511100100, and in part by the Shanghai Rising-Star Program under Grant 20QA1404600. *(Corresponding author: Chenglin Li.)*

References

[1] 2018 fgcvx fungi classification challenge. URL https://www.kaggle.com/competitions/fungi-challenge-fgvc-2018/overview.

[2] Sébastien M. R. Arnold, Guneet S. Dhillon, Avinash Ravichandran, and Stefano Soatto. Uniform sampling over episode difficulty. *CoRR*, abs/2108.01662, 2021. URL https://arxiv.org/abs/2108.01662.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[4] Daniel A Braun, Carsten Mehring, and Daniel M Wolpert. Structure learning in action. *Behavioural brain research*, 206(2):157–165, 2010.

[5] Zachary Charles and Jakub Konečný. On the outsized importance of learning rates in local update methods, 2020. URL https://arxiv.org/abs/2007.00878.

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017. URL https://arxiv.org/abs/1703.03400.

[8] Paul Glasserman. *Monte Carlo methods in financial engineering*. Springer, New York, 2004. ISBN 0387004513 9780387004518 1441918221 9781441918222.

[9] Allan Jabri, Kyle Hsu, Abhishek Gupta, Ben Eysenbach, Sergey Levine, and Chelsea Finn. Unsupervised curricula for visual meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] Jean Kaddour, Steindór Sæmundsson, et al. Probabilistic active meta-learning. *Advances in Neural Information Processing Systems*, 33:20813–20822, 2020.

[11] Xiaomeng Li, Lequan Yu, Yueming Jin, Chi-Wing Fu, Lei Xing, and Pheng-Ann Heng. Difficulty-aware meta-learning for rare disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 357–366. Springer, 2020.

[12] Chenghao Liu, Zhihao Wang, Doyen Sahoo, Yuan Fang, Kun Zhang, and Steven CH Hoi. Adaptive task sampling for meta-learning. In *European Conference on Computer Vision*, pages 752–769. Springer, 2020.

[13] Ricardo Luna Gutierrez and Matteo Leonetti. Information-theoretic task selection for meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 20532–20542, 2020.

[14] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

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

[16] Michael JD Powell and J Swann. Weighted uniform sampling—a monte carlo technique for reducing variance. *IMA Journal of Applied Mathematics*, 2(3):228–236, 1966.

[17] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

[18] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017. URL https://arxiv.org/abs/1703.05175.

[19] Philip S. Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning, 2016. URL https://arxiv.org/abs/1604.00923.

[20] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2016. URL https://arxiv.org/abs/1606.04080.

[21] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[22] Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, pages 7045–7054. PMLR, 2019.

[23] Huaxiu Yao, Yu Wang, Ying Wei, Peilin Zhao, Mehrdad Mahdavi, Defu Lian, and Chelsea Finn. Meta-learning with an adaptive task scheduler. *CoRR*, abs/2110.14057, 2021. URL https://arxiv.org/abs/2110.14057.