Variance-reduced First-order Meta-learning for Natural Language Processing Tasks

Lingxiao Wang[1], **Kevin Huang**[2], **Tengyu Ma**[3], **Quanquan Gu**[1], **Jing Huang**[2]
[1]Department of Computer Science, University of California, Los Angeles
[2]JD AI Research, Mountain View, CA
[3]Department of Computer Science, Stanford University
{lingxw,qgu}@cs.ucla.edu
{jing.huang,kevin.huang3}@jd.com
tengyuma@stanford.edu

Abstract

First-order meta-learning algorithms have been widely used in practice to learn initial model parameters that can be quickly adapted to new tasks due to their efficiency and effectiveness. However, existing studies find that meta-learner can overfit to some specific adaptation when we have heterogeneous tasks, leading to significantly degraded performance. In Natural Language Processing (NLP) applications, datasets are often diverse and each task has its unique characteristics. Therefore, to address the overfitting issue when applying first-order meta-learning to NLP applications, we propose to reduce the variance of the gradient estimator used in task adaptation. To this end, we develop a variance-reduced first-order meta-learning algorithm. The core of our algorithm is to introduce a novel variance reduction term to the gradient estimation when performing the task adaptation. Experiments on two NLP applications: few-shot text classification and multi-domain dialog state tracking demonstrate the superior performance of our proposed method.

1 Introduction

Meta-learning has recently emerged as a promising approach in solving many natural language processing tasks, such as few-shot text classification (Obamuyide and Vlachos, 2019; Bao et al., 2019), low resource language understanding (Gu et al., 2018; Dou et al., 2019; Yu et al., 2020a), and multi-domain dialogue systems (Qian and Yu, 2019; Huang et al., 2020). In particular, model-agnostic meta-learning (MAML) (Finn et al., 2017), a widely-used meta-learning approach, trains an initial model that can be adapted to a new task with a small number of optimization steps and training data. However, MAML requires the computation of second-order derivatives, which can be costly for reinforcement learning and NLP applications. Therefore, numerous computationally-efficient MAML variants (Finn et al., 2017; Li et al., 2017; Nichol et al., 2018; Antoniou et al., 2018; Zintgraf et al., 2019; Song et al., 2020) have been proposed in recent years. First-order meta-learning (Finn et al., 2017; Nichol et al., 2018) is a widely-used method in practice because it is easy to implement, eliminates computationally-intensive second-order derivatives in MAML, and achieves state-of-the-art performance.

Although meta-learning including first-order meta-learning has shown promising performances in many applications (Triantafillou et al., 2019), it still somewhat struggles to learn on diverse task distributions (Triantafillou et al., 2020; Rebuffi et al., 2017; Yu et al., 2020c). For first-order meta-learning, it consists of task adaptation and meta updates. Task adaptation aims to obtain a task-specific model for each task by performing several optimization steps based on the current meta model. Then, the meta update aggregates the gradient information of task-specific models to obtain a new meta model. It has been observed in many previous works (Zhao et al., 2018; Karimireddy et al., 2019; Charles and Konečnỳ, 2020) that local update methods, including first-order meta-learning, performing multiple optimization steps on local data can lead to overfitting to atypical local data. In the context of first-order meta-learning, due to the large variance of the gradient estimator, task adaptation will drive task-specific models to move away from each other, resulting in that the gradients used in meta update have diverse directions. Furthermore, since the difference in gradient magnitudes will also be large, the task with a much larger gradient in magnitude will dominate the task adaptation. As a result, the meta update will overfit to this dominating task. Similar issues have been studied in multi-task learning: Yu et al. (2020b) showed that conflicting gradients, i.e., two gradients that have a negative cosine similarity, can lead to significantly degraded performance when the difference in gradient magnitudes is large.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2609–2615
June 6–11, 2021.

The above gradient variance issue, i.e., the large variance from the gradient estimator, is significant in NLP applications since many NLP datasets have diverse properties, and the tasks for meta-learning in NLP applications also have their unique characteristics. For example, the MultiWOZ dataset (Budzianowski et al., 2018) for dialog systems and the Spider dataset (Yu et al., 2018) for semantic parsing, both consist of complex and cross domain examples. To address the aforementioned gradient variance issue in NLP applications when applying first-order meta-learning approaches, we propose a variance-reduced first-order meta-learning (VFML) algorithm. The key idea of our algorithm is that we leverage a novel variance reduction term in the task adaptation steps to reduce the variance of the gradient estimator. We evaluate our proposed method on two NLP applications: few-shot text classification and domain adaptation in multi-domain dialog state tracking. We experiment on several benchmark datasets, finding that our method produces models that can achieve better performances than the baseline Reptile (Nichol et al., 2018).

2 Problem Setup and Preliminaries

Let $\mathcal{T} = \{\mathcal{T}_i\}_{i\in\mathcal{I}}$ be the set of all tasks and $\mathcal{I}$ be the task index set. Suppose $\mathcal{T}_i$ is drawn from $\mathcal{T}$ with probability p_i, and we use p to denote the probability distribution over $\mathcal{T}$. Our goal is to find an initial model $\boldsymbol{\theta}$ such that it will have a small loss on a new task $\mathcal{T}_i$ after a few steps of updates. Therefore, we want to solve the following problem

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \mathbb{E}_{i\sim p}[L_i(f_i^K(\boldsymbol{\theta}))], \qquad (2.1)$$

where $f_i^K(\boldsymbol{\theta})$ is the function that updates the initial model parameter $\boldsymbol{\theta}$ for K steps on task $\mathcal{T}_i$.

2.1 First-order meta learning

To solve the problem in equation 2.1, MAML uses task adaptation, i.e., $f_i^K(\boldsymbol{\theta})$, and the following meta update based on sampled tasks

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \tau \textstyle\sum_{i\in\mathcal{I}_b} \nabla L_i(f_i^K(\boldsymbol{\theta}))/|\mathcal{I}_b|,$$

where τ is the step size, $\mathcal{I}_b$ is the index set of the sampled tasks, and $f_i^K(\boldsymbol{\theta})$ is usually K steps of gradient descent. A more efficient and effective MAML variant is the first-order method (Finn et al., 2017; Nichol et al., 2018). For instance, Finn et al. (2017) proposed to replace the Hessian matrix in meta update with an identity matrix, which leads to First-order MAML (FOMAML). Nichol et al. (2018) proposed Reptile to further simplify FOMAML by using the the following meta update

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \tau \textstyle\sum_{i\in\mathcal{I}_b} (\boldsymbol{\theta}_i' - \boldsymbol{\theta})/|\mathcal{I}_b|,$$

where $\boldsymbol{\theta}_i' = f_i^K(\boldsymbol{\theta})$. In this work, we propose a new method based on Reptile to improve the performance of first-order meta-learning methods.

3 Method

Our proposed algorithm for meta-learning is illustrated in Algorithm 1. In the following discussion, we use $\nabla L_{i,\mathcal{B}_t^i}$ to denote the mini-batch stochastic gradient for task i and $\mathcal{B}_t^i$ is the sample index set. The main idea of our method is to construct

Algorithm 1 Variance-reduced First-order Meta-learning (VFML) Algorithm

input initialization $\boldsymbol{\theta}_0$, initial variance reduction term $\mathbf{v}_0$, step size: η, τ, iteration numbers: T, K, parameters: β, γ

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: Sample Tasks $\mathcal{I}_t \subseteq \mathcal{I}$ with $|\mathcal{I}_t| = m$
3: **for** $i \in \mathcal{I}_t$ **do**
4: $\mathbf{w}^i = \text{Task Adaptation}(\boldsymbol{\theta}_t, \mathbf{v}_t, \eta, K, \gamma, i)$
5: **end for**
6: Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \tau \frac{1}{m} \sum_{i\in\mathcal{I}_t} \left(\mathbf{w}^i - \boldsymbol{\theta}_t\right)$
7: Update $\mathbf{v}_{t+1} = \frac{1}{m} \sum_{i\in\mathcal{I}_t} \nabla L_{i,\mathcal{B}_t^i}(\boldsymbol{\theta}_{t+1}) + (1-\beta)\left(\mathbf{v}_t - \frac{1}{m} \sum_{i\in\mathcal{I}_t} \nabla L_{i,\mathcal{B}_t^i}(\boldsymbol{\theta}_t)\right)$
8: **end for**

output $\boldsymbol{\theta}_T$

a variance reduction term $\mathbf{v}$, which is motivated by the stochastic recursive momentum technique proposed in (Cutkosky and Orabona, 2019). $\mathbf{v}$ will be used in the task adaptation step (line 4 in Algorithm 1) to reduce the variance of the gradient estimator. More specifically, we use the gradient estimator $\mathbf{g}_k^i = \gamma \nabla L_{i,\mathcal{B}_k^i}(\mathbf{w}_k^i) + (1-\gamma)\mathbf{v}$ (line 3 in Algorithm 2) to update the task-specific model for task $\mathcal{T}_i$. $\mathbf{g}_k^i$ is a weighted sum of the mini-batch stochastic gradient $\nabla L_{i,\mathcal{B}_k^i}(\mathbf{w}_k^i)$ and the variance reduction term $\mathbf{v}$, and $(1-\gamma)$ is the weight for $\mathbf{v}$. When $\gamma = 1$, it reduces to Reptile. We initialize the variance reduction term $\mathbf{v}_0$ by averaging the gradients from a set of tasks which are randomly sampled and computed using the initialization $\boldsymbol{\theta}_0$.

Next, we briefly discuss the intuition of why our proposed method can reduce the variance of the gradient estimator. Suppose $\mathbb{E}\left\|\nabla L_{i,\mathcal{B}_k^i}(\boldsymbol{\theta}) -\right.$

Algorithm 2 Task Adaptation (TA)

input meta model $\boldsymbol{\theta}$, variance reduction term $\mathbf{v}$, step size η, iteration number K, task index i, γ

1: $\mathbf{w}_0^i = \boldsymbol{\theta}$
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: $\quad \mathbf{g}_k^i = \gamma \nabla L_{i,\mathcal{B}_k^i}(\mathbf{w}_k^i) + (1-\gamma)\mathbf{v}$
4: $\quad \mathbf{w}_{k+1}^i = \mathbf{w}_k^i - \eta \mathbf{g}_k^i$
5: **end for**

output $\mathbf{w}_K^i$

$\nabla L_i(\boldsymbol{\theta})\big\|_2^2 \leq \sigma_1^2$ and $\mathbb{E}\big\|\nabla L_i(\boldsymbol{\theta}) - \nabla L(\boldsymbol{\theta})\big\|_2^2 \leq \sigma_2^2$, where $L(\boldsymbol{\theta}) = \mathbb{E}[L_i(\boldsymbol{\theta})]$. σ_1^2 is the variance of using $\nabla L_{i,\mathcal{B}_k^i}$ to estimate the gradient ∇L_i for task $\mathcal{T}_i$. σ_2^2 is the variance introduced by the dissimilarity between tasks. Intuitively, the variance of the gradient estimator in Reptile, i.e., $\mathbb{E}\big\|\nabla L_{i,\mathcal{B}_k^i}(\mathbf{w}_k^i) - \nabla L(\mathbf{w}_k^i)\big\|_2^2$, will be determined by the following quantity

$$O(\sigma_1^2 + \sigma_2^2).$$

In addition, the variance of the gradient estimator in VFML, i.e., $\mathbb{E}\big\|\mathbf{g}_k^i - \nabla L(\mathbf{w}_k^i)\big\|_2^2$, will be determined by

$$O\big(\sigma_1^2 + \gamma^2\sigma_2^2 + (1-\gamma)^2(\beta^2\sigma_2'^2 + (1-\beta)^2\boldsymbol{\Delta}_{t+1}^2)\big),$$

where $\boldsymbol{\Delta}_{t+1}^2 = \mathbb{E}\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|_2^2$, $\sigma_2'^2 = \mathbb{E}\big\|\sum_{i\in\mathcal{I}_t}\nabla L_i(\boldsymbol{\theta}_t)/m - \nabla L(\boldsymbol{\theta}_t)\big\|_2^2$.

If we have a large number of examples for each task, then σ_1^2 will be small, and the variance of the gradient estimator in Reptile will be determined by $O(\sigma_2^2)$. When we have very diverse task distributions, σ_2^2 will be large, which can lead to a significant degradation in performance. However, for VFML, the variance will be dominated by $O\big(\gamma^2\sigma_2^2 + (1-\gamma)^2(\beta^2\sigma_2'^2 + (1-\beta)^2\boldsymbol{\Delta}_{t+1}^2)\big)$. Since $\sigma_2'^2$ can be much smaller than σ_2^2 and $\boldsymbol{\Delta}_{t+1}^2$ goes to zero as our algorithm convergences, the variance of $\mathbf{g}_k^i$ can be much smaller than σ_2^2 by choosing appropriate parameters β, γ. Therefore, the role of the variance reduction term $\mathbf{v}$ is to alleviate the variance introduced by the task dissimilarity.

4 Experiments

We evaluate our proposed method on one simulation experiment and two NLP applications: text classification and dialog state tracking.

4.1 Simulation

To validate the effectiveness of our proposed method, we consider the one-dimensional sine wave regression (Finn et al., 2017; Nichol et al., 2018). Our goal is to learn a neural network that can quickly adapt to a given sine wave function after a few adaptation steps. We follow the same experimental setup in the previous work (Nichol et al., 2018), and we compare our proposed method with Reptile (Nichol et al., 2018) in terms of the mean square error between the output of the adapted neural network and the sine wave function.

Parameters: For both methods, we sample 10 tasks at each outer loop iteration and use 10 examples, i.e., $b = 10$, to compute the mini-batch stochastic gradients. We choose $K = 3$, $\eta = 0.01$ for the task adaptation step, and choose $\tau = 1$ for the meta update. For our proposed method, we choose γ by searching the grid $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and β by $\{0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$.

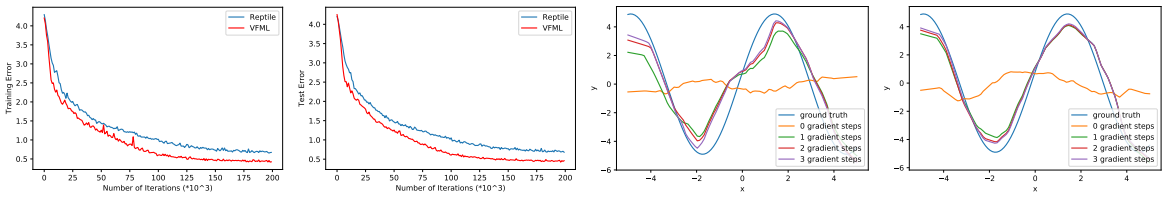
Results: Figures 1(a) and 1(b) shows the training and test accuracy versus the number of iterations for our method and Reptile. Figures 1(c) and 1(d) illustrate the adaptation results for both methods. Figures 1(a) and 1(b) show that VFML can reduce the iteration numbers and achieve better performance in terms of training and test accuracy than Reptile. Figures 1(c) and 1(d) illustrates that our proposed method can quickly converge to a given sine wave function. These results validate the superiority of VFML.

4.2 Few-shot Text Classification

We consider two text classification datasets: *Amazon* (He and McAuley, 2016) and *FewRel* (Han et al., 2018). For *Amazon* dataset, it consists of customer reviews from 24 product categories, and we follow the previous work (Bao et al., 2019) to sample 1000 reviews for each category. For this dataset, our goal is to classify a given review into its corresponding product category. *FewRel* is a relation classification dataset, and each example is a sentence annotated with a head entity, a tail entity, and their relation. For *FewRel*, we aim to predict the relation between the head and tail in a given sentence.

We follow the experimental setup in previous work (Bao et al., 2019). We consider the N-way K-shot setting, where N is the number of classes in each task, and K is the number of examples in the class.

Baseline models: For this problem, we consider the convolutional neural network (CNN) based



(a) Training Error (b) Test Error (c) Model by Reptile (d) Model by VFML

Figure 1: Results on one-dimensional sine wave regression. Figures 1(a) and 1(b) show the training error and test error versus the number of iterations. Figures 1(c) and 1(d) demonstrate the adaptation results of different methods.

Method	*Amazon*	*FewRel*	*Amazon*	*FewRel*	*Amazon*	*FewRel*
	5-shot	5-shot	10-shot	10-shot	50-shot	50-shot
Reptile	66.6 ± 1.92	67.6 ± 0.78	72.5 ± 2.65	72.9 ± 1.50	61.88 ± 3.91	72.2 ± 1.61
Ours	$\mathbf{67.6 \pm 1.40}$	$\mathbf{68.4 \pm 0.61}$	$\mathbf{74.8 \pm 1.33}$	$\mathbf{74.00 \pm 0.68}$	$\mathbf{63.90 \pm 4.72}$	$\mathbf{74.7 \pm 0.89}$

Table 1: Results of text classification on *Amazon* and *FewRel* datasets. We consider 5-way N-shot settings with $N = 5, 10, 50$. We report the classification accuracy with the standard deviation over 10 trials.

5-shot	MAML	FO-MAML	Reptile	**Ours**
Amazon	63.3 ± 1.87	65.8 ± 1.63	66.6 ± 1.92	$\mathbf{67.6 \pm 1.40}$
FewRel	67.7 ± 0.73	66.9 ± 2.16	67.6 ± 0.78	$\mathbf{68.4 \pm 0.61}$

Table 2: Results of different meta-learning methods on *Amazon* and *FewRel* datasets in 5-way 5-shot settings. We report the classification accuracy with the standard deviation over 10 trials.

Method	*Taxi* (1%)		*Attrac* (1%)		*Taxi* (5%)		*Attrac* (5%)		*Taxi* (10%)		*Attrac* (10%)	
	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot	Joint	Slot
Train from scratch	60.52	72.90	27.88	63.43	60.52	72.90	43.15	73.27	60.52	72.90	50.16	78.09
Reptile	60.91	76.10	40.71	73.01	61.67	82.40	51.38	78.54	63.94	84.94	53.12	80.13
Ours	**62.00**	**77.45**	**43.15**	**74.02**	**65.66**	**84.32**	**52.35**	**79.59**	**67.16**	**86.03**	**56.14**	**81.13**

Table 3: Results on DST. We report the joint and slot accuracy for different methods under different number of finetune examples. 1% means that we use 1% of the new domain data for training from scratch and finetune.

model proposed in (Bao et al., 2019). More specifically, we use a CNN as the embedding model to generate the input representation and a one-hidden-layer neural network with 300 units and ReLU activation as the classifier.

Parameters: For both Reptile and our method, we choose K by searching the grid $\{1, 3, 5, 10\}$, η by $\{0.01, 0.05, 0.1, 0.3, 0.5\}$ for the task adaptation step, and choose $\tau = 1$ for the meta update. For our proposed method, we choose γ by searching the grid $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and β by $\{0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$.

Results: Table 1 summarizes the comparisons of different methods on *Amazon* and *FewRel* datasets for text classification. The results are averaged over 10 runs. In the 5-way 5-shot setting, our proposed method can achieve 1% and 0.8% improvements in terms of classification accuracy on *Amazon* and *FewRel* datasets, respectively.

Analysis: We also consider the 5-way 10-shot and 5-way 50-shot settings. These two settings are used to evaluate our proposed method's performance when the variance of the gradient estimator is dominated by the variance introduced by the task dissimilarity. The results show that, when we have 50 shots, our proposed method can achieve 2.02% and 2.5% gains on classification accuracy on *Amazon* and *FewRel* datasets, respectively. The results in 10-shot and 50-shot settings validate the effectiveness of the variance reduction term, i.e., it is used to alleviate the variance of the gradient estimator introduced by the task dissimilarity. We also compare our proposed method with the MAML and FO-MAML methods proposed in (Finn et al., 2017) on *Amazon* and *FewRel* datasets in 5-way 5-shot settings. Table 2 shows that our method outperforms these two baselines.

4.3 Dialog State Tracking

We also test our VRML method on the task of multi-domain dialog state tracking (DST). We experiment on the MultiWOZ (Budzianowski et al., 2018), a large scale, multi-domain human-human dialog state tracking dataset. It had been introduced to help facilitate research to solve the DST problem. This corpus contains 8438 multi-turn dialogues with on average of 13.7 turns per dialogue. Multi-domain dialog state tracking in MultiWOZ is a challenging task for meta-learning, due to the differences in dialogues between each domain. For example, the dialog states, and user utterances for hotel and train are quite different. We use the most frequent five domains: (*restaurant*, *hotel*, *attraction*, *taxi*, *train*). We follow the same setup in (Huang et al., 2020) by training on three source domains: *hotel*, *restaurant* and *train*, and testing on 1% of the target domains: (*taxi*, *attraction*).

We compare our method with Reptile and the train-from-scratch, i.e., we train a randomly initialized model using data from the target domain. We use joint and slot accuracy (Wu et al., 2019) to evaluate different methods. Joint accuracy measures the accuracy of dialogue states, where a dialogue state is correctly predicted only if all the values for (*domain*, *slot*) pairs are correctly predicted. Slot accuracy measures the accuracy of each (*domain*, *slot*, *value*) tuples for the dialog state.

Baseline models: We quantify the benefits of different meta-learning algorithms by comparing the results on top of the TRADE model architecture (Wu et al., 2019). TRADE is an encoder-decoder model utilizing two BiGRUs to encode sequences of dialogue turns, and then generating corresponding (domain, slot, value) tuples. We set the hidden size of the encoder and decoder to be 400 and use Glove embedding (Pennington et al., 2014).

Parameters: For both Reptile and our method, we choose K by searching the grid $\{1, 3, 5\}$, η by $\{0.01, 0.05, 0.1\}$ for the task adaptation step, and choose $\tau = 1$ for the meta update. For our proposed method, we choose γ by searching the grid $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and β by $\{0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. Following the previous work (Wu et al., 2019; Huang et al., 2020), we set batch size to 32, dropout rate to 0.2. For the finetune step, we search the batch size by the grid $\{4, 8, 16, 32\}$ and setp size by $\{0.01, 0.05, 0.1\}$. We early stop the training of both methods when the validation accuracy converges.

Results: Table 3 reports the joint and slot accuracy for different methods. The results show that, when we have 1% of the target domain data for finetuning, our proposed method can achieve 2.44% and 1.01% improvements in slot and joint accuracy compared with Reptile for *Attraction*. Compared with train-from-scratch, we can obtain 15.27% and 10.59% gains in slot and joint accuracy. Similar improvements can be obtained for *Taxi*.

Analysis: We also consider the case when we have more target domain data for finetuning. Table 3 shows that the more target domain data we have, the more gains our method can obtain. For example, when we have 10% data for *Taxi*, our method can achieve 6.64%/13.13% improvements in slot/joint accuracy compared with train-from-scratch. Compared with Reptile, we can obtain 3.32%/1.09% gains in slot/joint accuracy. Note that there is no change of performance for the train-from-scratch method on 1%/5%/10% Taxi data, due to the small size of the Taxi dataset. If we train on the entire Taxi data, the joint/slot accuracy would be 75.61%/89.61%. These results show that meta-learning indeed helps when the target data is small, and VRML is very effective on using the small amount of target data compared to Reptile.

5 Conclusion

We propose a novel first-order meta-learning method to reduce the variance of the gradient estimator used in task adaptation for NLP tasks. We show in both few-shot text classification and DST that our method can achieve better performance than existing methods. It is interesting to further study domain adaptation methods built upon our new algorithm.

References

Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2018. How to train your maml. In *International Conference on Learning Representations*.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2019. Few-shot text classification with distributional signatures. *arXiv preprint arXiv:1908.06039*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Zachary Charles and Jakub Konečnỳ. 2020. On the outsized importance of learning rates in local update methods. *arXiv preprint arXiv:2007.00878*.

Ashok Cutkosky and Francesco Orabona. 2019. Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems*, pages 15236–15245.

Zi-Yi Dou, Keyi Yu, and Antonios Anastasopoulos. 2019. Investigating meta-learning algorithms for low-resource natural language understanding tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1192–1197.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Yi Huang, Junlan Feng, Min Hu, Xiaoting Wu, Xiaoyu Du, and Shuo Ma. 2020. Meta-reinforced multi-domain state generator for dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7109–7118.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2019. Scaffold: Stochastic controlled averaging for federated learning. *arXiv preprint arXiv:1910.06378*.

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

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

Abiola Obamuyide and Andreas Vlachos. 2019. Model-agnostic meta-learning for relation classification with limited supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5873–5879.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516.

Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, and Yunhao Tang. 2020. Es-maml: Simple hessian-free meta learning. In *International Conference on Learning Representations*.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. 2020. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. 2019. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Changlong Yu, Jialong Han, Haisong Zhang, and Wilfred Ng. 2020a. Hypernymy detection for low-resource languages via meta learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3656.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020b. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. 2020c. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. 2019. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR.