THE CHINESE UNIVERSITY OF HONG KONG, SHENZHEN

DDA4210: Advanced Machine Learning

Course Project

Gesture Recognition in Dynamic Environments using MAML-DG

Group 28
Wu Dongze 119010337
Ma Kexuan 120090651
Chen Zhilin 120090814
He Jingge 120090899

May 06, 2023

CONTENT

I. INTRODUCTION

Gesture recognition is a fundamental technology for a wide range of applications, including smart homes, security surveillance, and virtual reality. Traditional approaches rely on cameras, wearable devices, phones, or sonar for sensing. However, these approaches have limitations such as privacy leakage, the need for on-body sensors, and limited sensing range. To overcome these challenges, researchers have proposed sensing solutions based on Wi-Fi signals from commodity devices, such as E-eyes [1], CARM [2], WiGest [3], and WIMU [4].

Early Wi-Fi sensing solutions extract statistical or physical features from Wi-Fi signals and map them to human gestures. However, these primitive signal features may include irrelevant environmental information. In this paper, we refer to these irrelevant factors as "domain." As a result, classifiers trained with primitive signal features in one domain may suffer a drastic drop in accuracy when used in another domain. Therefore, we aim to build a "one-fits-all" model that can be trained once and used anywhere. This model should be able to generalize across different domains without extra efforts, such as data collection, generation, or re-training.

The most cutting-edge work in this field is completed by Tsinghua University. hey developed a dataset named Widar 3.0 [5] specifically for intelligent wireless sensing and utilizing Convolutional Neural Networks for gesture recognition. However, Widar does not account for the Wi-Fi signals interference from multiple sources Besides, simple CNN can not generalize the model well.

Therefore, our key idea is to move the generalization ability downward to the lower signal level, rather than the upper model level. Specifically, we extract domain-independent features that only reflect the gestures themselves from raw domain-dependent signals. Based on these features, we aim to build an explainable cross-domain recognition model named MAML-DG CNN that can be applied in new scenarios with zero effort and high accuracy.

II. SIGNIFICANCE AND NOVELTY

A. Mobility of Time

Wi-Fi signals are prone to interference from multiple sources, such as the number of devices connected to the network, the distance between the device and the router, and interference from other devices or networks. Widar 3.0, while being a widely used approach, does not account for these variations, which results in poor performance when tested on data from a different day. On the other hand, our proposed model is capable of accommodating different timings, leading to efficient performance on testing data collected at various times. The model's capability to adapt to diverse timings is highly beneficial for real-world applications, where certain essential characteristics persist across a sequence of time, reducing the influence of time variation.

B. Efficient Training in Diverse Domains

In the context of machine learning, the term "domain" refers to a distinct set of data with a specific distribution. While the Widar method defines different places as representing diverse domains, our approach considers different times as separate domains. The demo codes of the CNN algorithm used in the Widar 3.0 reveal that the performance of the model is sub-optimal when tested with data points from a new date. Conversely, our approach, which employs cross-time domains, enables more efficient training and testing on data from a new date. By leveraging this cross-time domain setting, our model can effectively adapt to new environments and perform well on diverse datasets.

C. Adapt Quickly to New Environments

Our model is able to to quickly adapt to environmental changes with computationally inexpensive updates sets it apart from the competition. Furthermore, MAML algorithm achieves this adaptation with just three CSI images per point in a new environment, solidifying its position as a cost-effective solution. This means that we can reduce the amount of data, which is beneficial when dealing with limited resources or when collecting data is challenging. The remainder of the report is structured as follows: Section III gives the data collection from Widar 3.0. The proposed algorithm is then presented in detail in Section IV. In Section V, we demonstrate the efficacy of the proposed scheme through simulation and real-world data, while the report is concluded in Section VI. For clarity, the notations adopted throughout the paper are summarized in Table I.

III. DATA COLLECTION AND PREPROSESSING

A. Channel State Information (CSI)

CSI is a vital concept in modern wireless communication systems. It refers to the information regarding the state of a communication channel between a transmitter and a receiver. The channel can be affected by various factors such as path loss, shadowing, fading, and interference. CSI can be used by the transmitter to adapt its transmission parameters such as power and modulation scheme to optimize the performance of the communication system. It is typically obtained by measuring the signal characteristics at the receiver, and then sending the measurements back to the transmitter for processing.

B. Body-coordinate Velocity Profile(BVP)

BVP is a measure of the velocity of an object in relation to its own frame of reference. In the context of human movement analysis, BVP refers to the speed and direction of movement of a body part relative to the rest of the body. It is a key parameter used in biomechanics to quantify human motion during various activities such as walking, running, and jumping. BVP can be measured using various motion capture systems such as optical, magnetic, or inertial sensors, and is typically reported in meters per second or degrees per second.

C. Widar Preprocessing

As shown in Fig. 1, to collect Channal State Information (CSI) data, multiple wireless links are deployed around the monitoring area. The wireless signals, distorted by the user

TABLE I: Important notations used throughout the paper

Notation	Description
$\mathcal{P}(\tau)$	The overall distribution of tasks
$\mathcal{P}_i(\tau)$	Distribution of tasks from domain i
$f_{\boldsymbol{\theta}}$	Model output with parameters $\boldsymbol{\theta}$
N	The number of classes in each task
k_{spt}	The number of samples of the support set in each task
k_{qry}	The number of samples of the query set in each task
$D^s_{\tau_i}$	Support set of localization task τ_i that contains k_{spt} number of samples under each location of N ways
$D^q_{\tau_i}$	Query set of localization task τ_i that contains k_{qry} number of samples under each location of N ways
$\mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}, D^s_{\tau_i})$	The task-specific loss function for task τ_i based on model parameters $\boldsymbol{\theta}$ and support set $D^s_{\tau_i}$
$\mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}, D^q_{\tau_i})$	The task-specific loss function for task τ_i based on model parameters $\boldsymbol{\theta}$ and query set $D^q_{\tau_i}$
$\boldsymbol{\theta}'_i$	The task-specific parameters after the inner loop via one step of gradient descent
$\boldsymbol{\theta}^*$	Updated meta-parameters after the outer loop
$\boldsymbol{\theta}_T(Q)$	The task-specific adapted model parameters obtained by updating $\boldsymbol{\theta}^*$ after Q steps of gradient descent
α	The step size of the inner loop
β	The step size of the outer loop

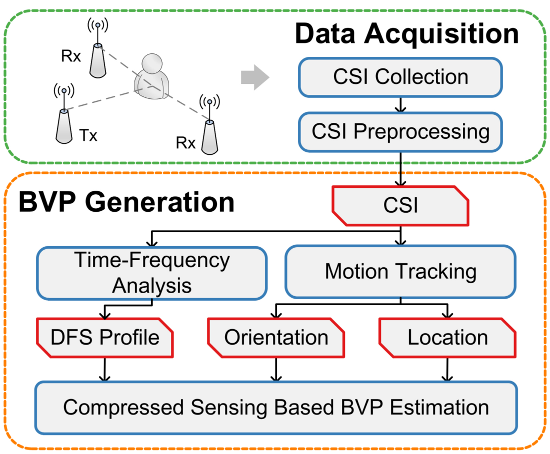


Fig. 1: Widar 3.0 preprocessing overview

in the monitoring area, are acquired at receivers, and their CSI measurements are logged and preprocessed to remove amplitude noises and phase offsets. Widar 3.0 divides the CSI series into small segments and generates a Body-coordinate Velocity Profile (BVP) for each segment through the BVP generation module. The system prepares three intermediate results: Doppler Frequency Shift (DFS) profiles, the orientation, and location information of the person. DFS profiles are estimated by applying time-frequency analysis to the CSI series, while the orientation and location information of the person are calculated using motion tracking approaches. Widar 3.0 then applies a compressed-sensing-based optimization approach to estimate the BVP of each CSI segment. The resulting BVP series (images) is then output for further gesture recognition.

IV. Algorithm

A. Model-Agnostic Meta-Learning (MAML)

Meta-learning is a learning to learn approach that enables the learning model to adapt to new tasks by leveraging previous experience from related tasks. In this framework, the training is based on the tasks. Tasks are drawn from a specific distribution, denoted as $\tau \sim \mathcal{P}(\tau)$, and each task includes a support set and a query set. In an N-way k_{spt}-shot k_{qry}-shot classification problem, a task consists of N classes, each with k_spt samples in the support set and k_qry samples in the query set. In the meta-training stage, M training tasks, $\{\tau_i\}_{i=1}^{M} \sim \mathcal{P}(\tau)$, are sampled from the distribution and the corresponding datasets are made available to the model. In the meta-testing stage, a new test task $T \sim \mathcal{P}(\tau)$ is presented, consisting of a small support set and a query set. The objective of meta-learning is to train a model on the M training tasks, such that it can quickly adapt to the new test task using the small support set and perform well on its query set.

Model-agnostic meta-learning (MAML) [6] does so by learning a set of initial parameters $\boldsymbol{\theta}_{MAML}$ for neural networks that enable good performance on a new task with only a few gradient descent steps. In the meta-training stage, MAML formulates a meta-optimization problem to find $\boldsymbol{\theta}_{MAML}$ as:

$$\boldsymbol{\theta}_{MAML} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{M} \mathcal{L}_{\tau_i}\left(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \hat{\mathcal{L}}_{\tau_i}(\boldsymbol{\theta})\right), \quad (1)$$

where it contains two task-specific loss functions $\hat{\mathcal{L}}_{\tau_i}$ and $\mathcal{L}_{\tau_i}$ computed based on the support set and query set of the training task τ_i, respectively. Then the meta-parameters are updated via stochastic gradient descent (SGD):

$$\boldsymbol{\theta}_{MAML} \leftarrow \boldsymbol{\theta}_{MAML} - \beta \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{M} \mathcal{L}_{\tau_i}\left(\boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \hat{\mathcal{L}}_{\tau_i}(\boldsymbol{\theta})\right), \quad (2)$$

where α and β denote the step size of the inner loop and outer loop, respectively. During the meta-test stage, the meta-parameters $\boldsymbol{\theta}_{MAML}$ are fine-tuned to obtain the parameters $\boldsymbol{\theta}_T$ for the neural network used in the test task T. This is achieved by updating the meta-parameters using the gradient of the loss function $\hat{\mathcal{L}_T}(\boldsymbol{\theta}_{MAML})$ computed based on the support set of the test task, as follows:

$$\boldsymbol{\theta}_T \leftarrow \boldsymbol{\theta}_{MAML} - \alpha \nabla_{\boldsymbol{\theta}} \hat{\mathcal{L}_T}(\boldsymbol{\theta}_{MAML}), \quad (3)$$

Algorithm 1 Vanilla MAML

Require:
$\mathcal{P}(\tau)$: distribution over tasks;
α: step size of the inner loop;
β: step size of the outer loop;
Meta-training Stage (in the historical environments):
Randomly initialize $\boldsymbol{\theta}$;
For ite in iterations do:
  Sample training tasks $\{\tau_i\}_{i=1}^{M} \sim \mathcal{P}(\tau)$;
  For each i in $\{1, 2, \dots, M\}$ do:
    $\boldsymbol{\theta}_i' = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}}; D_{\tau_i}^s)$;
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\tau_i} \mathcal{L}_{\tau_i}(f_{\boldsymbol{\theta}_i'}; D_{\tau_i}^q)$;
return $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$ when it converges.
Meta-test Stage (in the new environment):
Sample a test task $T \sim \mathcal{P}(\tau)$;
$\boldsymbol{\theta}_T \leftarrow \boldsymbol{\theta}^* - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_T(f_{\boldsymbol{\theta}^*}; D_T^s)$;
return $\boldsymbol{\theta}_T^* \leftarrow \boldsymbol{\theta}_T$ when it converges.

For the vanilla MAML algorithm, we only have a theoretical guarantee (an upper bound) [7] that the algorithm can converge on different domains, shown as follows:

Theorem 1. *Suppose $\mathcal{L}_T(f_{\boldsymbol{\theta}}, D_T^s)$ is G-Lipschitz continuous and W-smooth with respect to the parameters $\boldsymbol{\theta}$, and α satisfies $\alpha \leq \frac{1}{W}$. Setting $\rho = 1+2\alpha W$, then for any $T \sim \mathcal{P}(\tau)$ with $D_T^s = \{(x_i, y_i)\}_{i=1}^{k_{spt}} \sim T$, we have*

$$\begin{aligned} ER(\boldsymbol{\theta}_T^Q) &\leq \frac{2G^2(\rho^Q - 1)}{k_{spt} * W} + E_{T\sim\tau} E_{D_T^s}[\mathcal{L}_T(\boldsymbol{\theta}_T^Q; D_T^s) - \mathcal{L}_T(\boldsymbol{\theta}_T^*)] \\ &\leq \frac{2G^2(\rho^Q - 1)}{k_{spt} * W} + \frac{1}{2\alpha} E_{T\sim\tau}[||\boldsymbol{\theta}^* - \boldsymbol{\theta}_T^*||_2^2]. \end{aligned}$$

B. Model-Agnostic Meta-Learning with Domain Generalization (MAML-DG)

However, the above error bound is not tight and can not guarantee that the model parameters learned through MAML can adapt well to a new domain. We address this issue by proposing model-agnostic meta-learning with domain generalization (MAML-DG). Unlike the original MAML approach which takes all the domains together for training, MAML-DG treats each training domain distinctively and mimics the train-test domain shift during the training stage.

As outlined in Algorithm 2, MAML-DG is designed to train a deep learning model with parameters $\boldsymbol{\theta}$ across S training domains, which may have different statistical distributions but share the same label and input features space. During each meta-training iteration, MAML-DG randomly selects two training domains $D_I, D_I = 1, 2, \dots, S$ and $D_{II}, D_{II} = 1, 2, \dots, S, D_I \neq D_{II}$ and generates tasks in these two domains. The full steps are as follows.

Step ①: We virtually train a domain-specific meta-parameters $\boldsymbol{\theta}'$ on the tasks generated from the training domain D_I using the vanilla MAML algorithm. We derive the first domain-specific loss function as

$$F(\cdot) = \sum_{\tau_i^{(D_I)}} \mathcal{L}_{\tau_i^{(D_I)}}\left(f_{\boldsymbol{\theta}_i^{(D_I)}}\right) = \sum_{i=1}^{M} \mathcal{L}_{\tau_i^{(D_I)}}\left(f_{\boldsymbol{\theta} - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\tau_i^{(D_I)}}}\right). \tag{4}$$

Algorithm 2 MAML-DG

Require:
$\{\mathcal{P}^{(i)}(\tau)\}_{i=1}^{S}$: distributions over tasks in S domains;
α: step size of the inner loop;
β: step size of the outer loop;
w: weight of the loss function of the second training domain D_{II};
Meta-training Stage (in the historical environments):
Randomly initialize $\boldsymbol{\theta}$;
For ite in iterations do:
  Sample two training domains D_I and D_{II} uniformly from $\{1, 2, ..., S\}$;
  Sample tasks $\{\tau_i^{(D_I)}\}_{i=1}^{M} \sim \mathcal{P}^{(D_I)}(\tau)$ in domain D_I;
  For i in range (M) do:
    $\boldsymbol{\theta}_i^{(D_I)} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_i^{(D_I)}}(f_{\boldsymbol{\theta}}; D^s_{\tau_i^{(I)}})$;
  $\boldsymbol{\theta}' = \boldsymbol{\theta} - \beta \nabla_{\boldsymbol{\theta}} \sum_{\tau_i^{(D_I)}} \mathcal{L}_{\tau_i^{(D_I)}}(f_{\boldsymbol{\theta}_i^{(D_I)}}; D^q_{\tau_i^{(D_I)}})$;
  Sample tasks $\{\tau_j^{(D_{II})}\}_{j=1}^{M} \sim \mathcal{P}^{(D_{II})}(\tau)$ in D_{II};
  For j in range (M) do:
    $\boldsymbol{\theta}_j^{(D_{II})} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\tau_j^{(D_{II})}}(f_{\boldsymbol{\theta}'}; D^s_{\tau_j^{(D_{II})}})$;
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}' - w\beta \nabla_{\boldsymbol{\theta}} \sum_{\tau_j^{(D_{II})}} \mathcal{L}_{\tau_j^{(D_{II})}}(f_{\boldsymbol{\theta}_j^{(D_{II})}}; D^q_{\tau_j^{(D_{II})}})$;
return $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$ when it converges.
Meta-test Stage (in the new environment):
Sample a test task $T \sim \mathcal{P}(\tau)$;
$\boldsymbol{\theta}_T \leftarrow \boldsymbol{\theta}^* - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_T(f_{\boldsymbol{\theta}^*}; D_T^s)$;
return $\boldsymbol{\theta}_T^* \leftarrow \boldsymbol{\theta}_T$ when it converges.

Step ②: With the initialization $\boldsymbol{\theta}'$ obtained in the previous step, we derive a second domain-specific loss function for the tasks generated from the training domain D_{II} using vanilla MAML once again, which is shown as

$$G(\cdot) = \sum_{\tau_j^{(D_{II})}} \mathcal{L}_{\tau_j^{(D_{II})}}(f_{\boldsymbol{\theta}_j^{(D_{II})}}) = \sum_{j=1}^{M} \mathcal{L}_{\tau_j^{(D_{II})}}\left(f_{\boldsymbol{\theta}' - \alpha\nabla_{\boldsymbol{\theta}'}\mathcal{L}_{\tau_j^{(D_{II})}}}\right). \tag{5}$$

Step ③: We sum up the two domain-specific losses $F(\cdot)$ and $G(\cdot)$, and then update the meta-parameters $\boldsymbol{\theta}$, which emulates the real-time train-test domain shifts and helps the model generalize faster after a few iterations.

Since we add several different domains into the meta-learning stage, and we do fine-tuning on the testing dataset, MAML-DG can alleviate overfitting and achieve good generalization across different training domains compared to the vanilla MAML.

V. Experimental Evaluations

In this section, we will formulate human gesture recognition as a classification problem and verify the efficacy of MAML and MAML-DG based on the Widar dataset.

For each task in the experiment, we set the number of gestures $N = 3$, the number of support data $k_{spt} = 2$ and the number of query data $k_{qry} = 3$. A CNN, which includes four convolution layers, pooling layers and a fully connected layer, is set as our base neural network architecture. During the meta-training stage, the step size of the inner loop α and the outer loop β are set to 0.01 and 0.001, respectively. In addition, we set the number of gradient descent steps for the inner loop to 5. The dataset contains data collected on five days, four of which are used for training tasks (i.e., historical environments), and data collected on the other day are used for test tasks (i.e., new environments). We conducted the testing on the data from each date and finally reported the average accuracy.

A. Convergence of accuracy with respect to the number of training steps

During the meta-training stage, the test accuracy with respect to the number of iterative steps of the outer loop for MAML, MAML-DG and traditional CNN, are shown in Fig. 2. The traditional CNN in Fig. 2(a) is fine-tuned with only three samples per label on a new test date, which is exactly the same as how we fine-tune MAML and MAML-DG. And we also demonstrate the accuracy of the traditional CNN when it is fine-tuned with large amount of data (20 samples per label) on a new test date, which corresponds to Fig. 2(b). Overall, MAML and MAML-DG can achieve faster convergence and much higher accuracy than the traditional CNN if they are fine-tuned in a same way (Result A). Besides, MAML and MAML-DG are still advantageous even though the traditional CNN is fine-tuned with large amount of data (Result B). Moreover, MAML-DG can achieve even better accuracy than MAML. This is as expected since MAML-DG captures the domain differences from the various environments.

B. Convergence of accuracy with respect to the number of fine-tuning steps

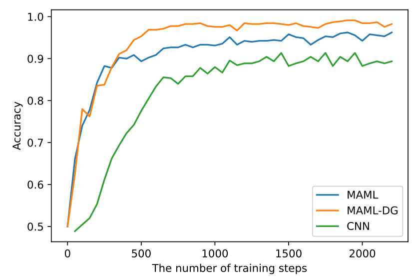
During the meta-test stage in a new environment, the well-trained meta-parameters are fine-tuned with small amount of data, and the convergence results of the test accuracy with respect to the number of fine-tuning steps are shown in Fig. 3. As demonstrated in the figure, MAML and MAML-DG only require 4 gradient steps during fine-tuning to achieve satisfactory testing accuracy. Besides, MAML-DG outperforms MAML in terms of even fewer fine-tuning steps and higher accuracy. This showcases the extraordinary environmental adaptation abilities of MAML and MAML-DG.

In summary, our proposed MAML and MAML-DG demonstrate exceptional environmental adaptation abilities, achieving fast adaptation on test tasks with only a small amount of data. Compared to the traditional CNN method, our algorithms offer superior performance, making it a highly promising choice for environmental adaptation.

VI. Conclusion

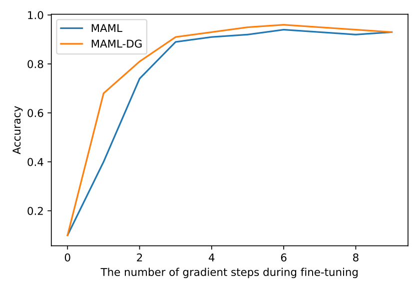
The MAML and MAML-DG framework proposed in this project provides a robust solution for wireless human gesture recognition in dynamic environments, ensuring fast adaptation and efficient performance. The framework includes two algorithms that leverage past knowledge to fine-tune meta-parameters, resulting in significant improvements in accuracy, convergence, and cost-effectiveness. Additionally, our proposed algorithm MAML-DG is more advantageous over MAML in terms of faster adaptation and higher accuracy. With its superior performance, MAML-DG has promising potential for many challenging 5G/6G scenarios such as urban canyons, underground parking, and rapid adaptation across different areas.

(a) Result A



(b) Result B

Fig. 2: Convergence comparison of the accuracy with respect to the number of training steps



(a) Hall

Fig. 3: Convergence comparison of the accuracy with respect to the number of gradient steps during fine-tuning

REFERENCES

[1] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures," *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, p. 617–628, 2014.

[2] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1118–1131, 2017.

[3] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1472–1480, 2015.

[4] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, p. 401–413, 2018.

[5] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3. 0: Zero-effort cross-domain gesture recognition with wi-fi," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44(11), pp. 8671–8688, 2021.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[7] P. Zhou, Y. Zou, X.-T. Yuan, J. Feng, C. Xiong, and S. Hoi, "Task similarity aware meta learning: Theory-inspired improvement on MAML," in *Proc. Uncertainty Artif. Intell.*, 2021, pp. 23–33.