Learning to Forget for Meta-Learning via Task-and-Layer-Wise Attenuation

Sungyong Baik, *Student Member, IEEE,* Junghoon Oh, *Student Member, IEEE,* Seokil Hong, *Student Member, IEEE,* and Kyoung Mu Lee, *Fellow, IEEE,*

Abstract—Few-shot learning is an emerging yet challenging problem in which the goal is to achieve generalization from only few examples. Meta-learning tackles few-shot learning via the learning of prior knowledge shared across tasks and using it to learn new tasks. One of the most representative meta-learning algorithms is the model-agnostic meta-learning (MAML), which formulates prior knowledge as a common initialization, a shared starting point from where a learner can quickly adapt to unseen tasks. However, forcibly sharing an initialization can lead to *conflicts* among tasks and the compromised (undesired by tasks) location on optimization landscape, thereby hindering task adaptation. Furthermore, the degree of conflict is observed to vary not only among the tasks but also among the layers of a neural network. Thus, we propose task-and-layer-wise attenuation on the compromised initialization to reduce its adverse influence on task adaptation. As attenuation dynamically controls (or selectively *forgets*) the influence of the compromised prior knowledge for a given task and each layer, we name our method Learn to Forget (L2F). Experimental results demonstrate that the proposed method greatly improves the performance of the state-of-the-art MAML-based frameworks across diverse domains: few-shot classification, cross-domain few-shot classification, regression, reinforcement learning, and visual tracking.

Index Terms—meta-learning, few-shot learning, MAML, reinforcement learning, visual tracking

1 Introduction

Deep learning models have recently demonstrated outstanding performance in various fields; however, they require supervised learning with a tremendous amount of labeled data, which entail a great amount of time and effort in preparing these data. On the other hand, humans are able to quickly learn new concepts from only few examples. Considering the considerable effort and time required to collect sufficient amounts of labeled data for each specific problem, the capability of humans to learn from few examples is desirable, especially for problems that require fast learning of new concepts.

When there exist concerns of overfitting in a few-data regime, data augmentation and regularization techniques are often used. Another commonly used technique is the fine-tuning of a network that is pre-trained on a large-scale dataset [1], [2]. Adaptation by fine-tuning often works without overfitting even in few-data regimes, but the computation cost is high due to the numerous update iterations [3]. By contrast, meta-learning can systematically approach the problem via two stages of learners: a meta-learner first learns common knowledge across a distribution of tasks, then the the learned common knowledge is used by a learner to quickly learn task-specific knowledge with few examples. One of the most popular instances is the model-agnostic meta-learning (MAML) [4], in which a meta-learner is formulated to learn a common initialization for encoding prior knowledge shared among tasks.

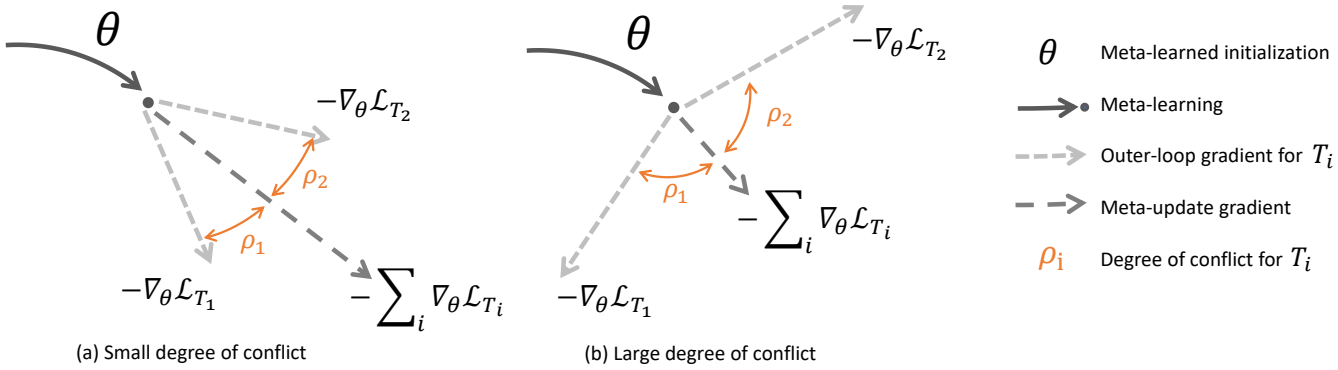
Although the assumption of the existence of a task distribution may justify the use of MAML for formulating a common initialization among tasks as a "good" starting point, variations among tasks still exist, some of which may lead to disagreements among tasks on the location of the initialization. We call such disagreement as *conflict* and formally define it in this study. Some of prior knowledge encoded in such a compromised initialization may be useful for one task but irrelevant or even detrimental for another task. Consequently, a learner will struggle to quickly learn new concepts when prior knowledge disagrees with the information from new examples. The learning difficulty can manifest as a sharp loss landscape, and thus, poor generalization [5], [6]. Motivated by our hypothesis, we analyze and observe the sharp landscape during fast adaptation to new examples (Figure 2) and suggest that the learned initialization by MAML is a "bad" starting point.

One solution for a meta-learner is to simply *forget* the conflicting part of prior knowledge that hinders the adaptation to a new task, thus minimizing its influence. This solution raises two questions: where do these *conflicts* occur, and to what extent? We hypothesize that the degree of conflict varies among the layers of a neural network, especially CNN, as deeper layers learn more task-specific knowledge or class-specific knowledge during classification [7]. To test the hypothesis, we measure *conflict* at each layer and have observed that *conflict* is indeed more severe at the deeper layers, as shown in Figure 3(a). We have also observed that the amount of agreement between the learned initialization and the initialization desired by a given task differs for each task, as shown in Figure 3(c).

Motivated by the aforementioned observations, we propose to learn selective *forgetting*[1] by applying a task-and-layer-wise *attenuation* on MAML initialization, in which

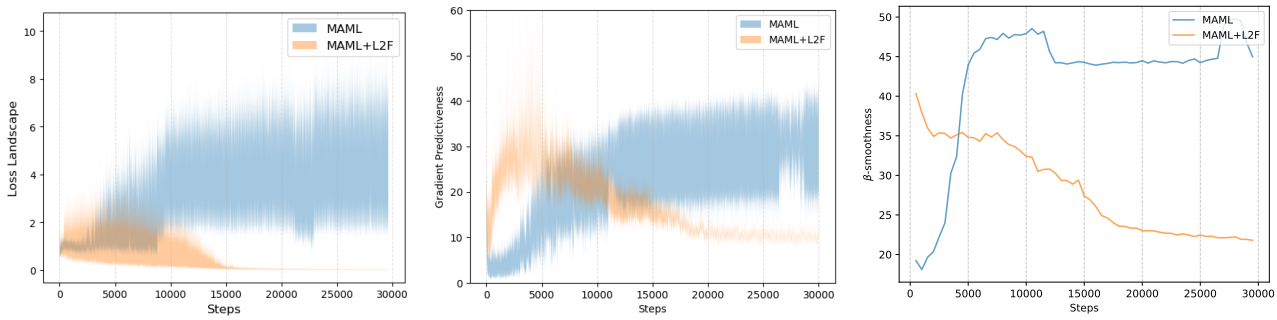
• *S. Baik, J. Oh, S. Hong, and K. M. Lee are with Automation and Systems Research Institute (ASRI), Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea.*
E-mail: {dsybaik, dh6dh, hongceo96, kyoungmu}@snu.ac.kr

1. The code is available at https://github.com/baiksung/L2F



(a) Small degree of conflict (b) Large degree of conflict

Fig. 1: **(a) A small degree of *conflict*:** When the desired initialization location for each task is close to each other, the overall meta-update gradient direction points towards the location that aligns well with the desired initialization for each task. **(b) A large degree of *conflict***, on the other hand, occurs when the direction of the desired initialization update for each task does not align. This scenario results in the overall meta-update gradient pointing towards the location desired by neither of tasks. A large degree of *conflict* is observed in MAML, especially at the higher layers of neural networks. Such undesired (hence compromised) initialization location can hinder learning during fast adaptation to each task, as illustrated in Figure 2.



(a) loss landscape (b) gradient predictiveness (c) "effective" β smoothness

Fig. 2: **Visualization of optimization landscape**. In [5], the stability and smoothness of the optimization landscape are analyzed by measuring Lipschitzness and the "effective" β-smoothness of loss. We use these measurements to analyze the learning dynamics of both MAML and our proposed method. Both approaches are trained on 5-way 5-shot miniImageNet classification tasks, *i.e.*, they are investigated for their inner-loop optimization. At each inner-loop update step, we measure the variations in loss (a), the l_2 difference in gradients (b), and the maximum difference in gradient over the distance (c) as we move to different points along the computed gradient for that gradient descent. We take an average of these values over the number of inner-loop updates and plot them against the training iterations. The thinner shade in plots (a) and (b) and the lower the values in plot (c) correspond to the smoother loss landscape, suggesting less training difficulty [5].

the level of the influence of prior knowledge is controlled for each task and layer. For each task, we argue that the initialization weights and its gradients, which are obtained from the support examples of a task, encode information about optimization specific to a task; thus we propose to condition on them to generate attenuation parameters. As for layer-wise attenuation, we generate an attenuation parameter for each layer. The proposed method, named Learn to Forget (L2F), indeed improves the quality of the initialization, as illustrated by the smoother loss landscape in Figure 2, and offers consistent performance improvement across different domains while managing to maintain the simplicity and generalizability of MAML. We extend our previous work [8] by reinforcing the generalizability of L2F by conducting additional experiments on large-scale few-shot classification, cross-domain few-shot classification, and visual tracking.

2 Related Work

The goal of meta-learning is to learn prior knowledge shared across tasks to achieve fast adaptation to new tasks [9], [10], [11], [12], [13]. Recent meta-learning systems can be classified into three categories: metric-based, network-based, and optimization-based systems. The metric-based system learns an embedding space, in which similar classes are closer whereas different classes are further apart [14], [15], [16], [17], [18]. Network-based approaches encode fast adaptation into the network architecture, such as by generating input-conditioned weights [19] or employing an external memory [20], [21]. On the other hand, optimization-based systems adjust optimization for fast adaptation [4], [22], [23].

Among optimization-based systems, MAML [4] has recently received a great amount of attention, owing to its simplicity and generalizability. The generalizability stems from its model-agnostic algorithm that learns across-task initialization. The initialization aims to encode prior knowledge that helps the model to quickly learn and achieve good generalization performance over tasks on the average. However, while MAML boasts simplicity, it has a relatively low performance on few-shot learning.

Recent works attribute the low performance to the ineffectiveness in fine-tuning feature extractors [24], meta-level overfitting [25], [26], [27], naïve update rules during fast adaptation [28], [29], [30], or the shared initialization [31], [32], [33]. Raghu *et al.* [24] observed that most of the performance of MAML comes from fine-tuning the classifier, claiming that fine-tuning the feature extractor is unnecessary. Similarly, recent works focused on improving the classifier, substantially boosting the performance of gradient-based meta-learning either by fine-tuning the linear classifier layer via convex optimization [34] or by initializing the classifier with the Prototypical Network-equivalent weights and bias [3]. Meanwhile, another set of research has attempted to reduce meta-level overfitting by adopting the information theory-based regularization objective [25], [26]. Another line of works has developed more advanced update rules, instead of naïve SGD with a fixed learning rate, for fast adaptation either by meta-learning a preconditioning matrix that preconditions gradients [30], [35] or meta-learning learning rates [28], [29]. Lastly, other studies have improved the performance of MAML by enabling the initialization to become task adaptive via affine transformation [31], weight generation [32], or hierarchical clustering of initializations [33].

By contrast, we approach the problem from the perspective of optimization and provide a new insight in which the quality of MAML initialization is compromised due to *conflicts* among tasks on the location of the initialization in optimization landscape. Such compromised initialization will hinder fast adaptation, as illustrated by the sharp loss landscape in Figure 2. Motivated by the phenomenon of *conflicts*, we argue that we only need to attenuate (*forget*) the compromised part of the initialization, thus leading to considerable performance boost (see Table 5).

Few concurrent works [36], [37] also attempt to address gradient conflicts, however in the context of multi-task learning. Chen *et al.* [36] attempt to address gradient conflicts by sampling gradients based on how consistent they are with each other, whereas Yu *et al.* [37] reduce gradient conflicts by projecting the conflicting gradients on to the normal planes of each other. In contrast to these works, we tackle gradient conflicts that occur in meta-learning (or MAML specifically) and thus can utilize meta-learning to meta-learn a network that can dynamically handle gradient conflicts.

Overall, L2F greatly improves MAML performance while maintaining its simplicity and generalizability. Owing to its generalizability, not only does L2F demonstrate a consistent improvement across diverse domains, but it can also be easily applied to improve other MAML-based methods, such as LEO [32], Warp-MAML [35], and Proto-MAML [3].

3 Proposed Method

3.1 Problem Formulation

Before discussing the proposed method in detail, this section introduces the formulation of a generic meta-learning algorithm. A general assumption is that there exist a distribution of tasks $p(\mathcal{T})$, from which a meta-learning algorithm aims to learn prior knowledge, as represented by a model with parameters θ. Tasks, which are sampled from $p(\mathcal{T})$, are split into three disjoint sets: meta-training, meta-validation, and meta-test sets. During the meta-training for k-shot learning, k number of examples are sampled from each task $\mathcal{T}_i$, which is sampled from the meta-training set. These k examples are then used to quickly adapt a base learner model with parameters θ', to the task $\mathcal{T}_i$. These examples are often called support examples and labeled as $\mathcal{D}^S_{\mathcal{T}_i}$ in this study. Then, new examples, which are disjoint from the support examples $\mathcal{D}^S_{\mathcal{T}_i}$, are sampled from the same task $\mathcal{T}_i$ to evaluate the generalization performance on unseen examples with the corresponding loss function $\mathcal{L}_{\mathcal{T}_i}$. These new unseen examples are often called query examples and are labeled as $\mathcal{D}^S_{\mathcal{T}_i}$ in this study. The feedback from the loss $\mathcal{L}_{\mathcal{T}_i}$ with query examples is then used to adjust the meta-learner model parameters θ in view of achieving better generalization. In particular, the feedback from the loss allows the meta-learner model to learn a better learning algorithm for the base learner model. Then, the base learner uses this learning algorithm to adapt to a new task, by using support examples, such that it performs well on new unseen examples, without overfitting. Finally, the meta-validation set and the meta-test set are used for the model selection and the final evaluation on the selected model, respectively. The processes as to how each task is sampled, how the disjoint set of support and query examples is sampled from each task, how the support examples are used for fast adaptation to a task, and how the query examples are used for the generalization evaluation are the same as those in the meta-training.

3.2 Model-Agnostic Meta-Learning

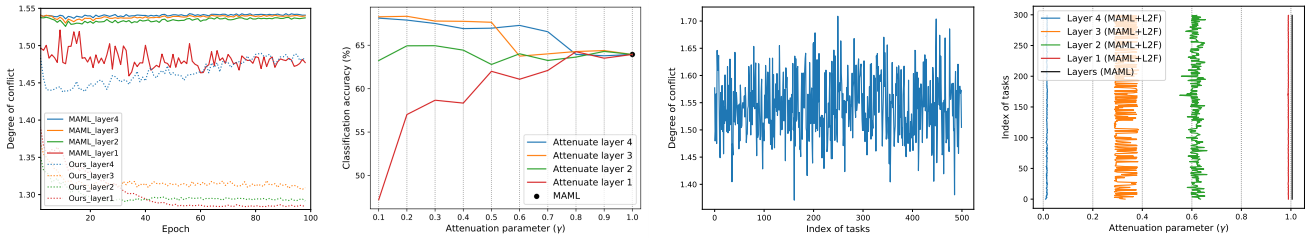
To achieve fast adaptation to new unseen tasks with few given examples, we borrow the philosophy and the methodology from MAML [4]. MAML formulates prior knowledge as a learnable initialization and thus seeks for a shared "good" initialization (initial set of values for weights) of a neural network across tasks. Formally, given a network f_θ parameterized by θ, MAML learns a set of initial weight values, θ, that will serve as a meta-learner or a good starting point for a base-learner to quickly adapt to a new task $\mathcal{T}_i$, sampled from a task distribution $p(\mathcal{T})$. In terms of the training algorithm, the whole process can be described as bi-level optimization: inner-loop optimization (or fast adaptation) and outer-loop optimization (or meta-update). For the inner-loop optimization, the network weights, as initialized by θ, are adapted to the task $\mathcal{T}_i$ by using few support examples $\mathcal{D}^S_{\mathcal{T}_i}$ and a task-specific loss function $\mathcal{L}_{\mathcal{T}_i}$ from $\mathcal{T}_i$ as follows:

$$\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^S_{\mathcal{T}_i}}(f_\theta), \tag{1}$$

where α is an inner-loop learning rate. For the outer-loop optimization, the meta-learner (or the learnable initialization θ) is given a feedback on the generalization performance of the model with adapted weights θ'_i to each task. To this end, the adapted model $f_{\theta'}$ can be evaluated on new query examples $\mathcal{D}^Q_{\mathcal{T}_i}$ sampled from the same task $\mathcal{T}_i$. The feedback, manifested in the form of loss gradients, is used to update the initialization θ such that better generalization can be achieved as follows:

$$\theta \leftarrow \theta - \eta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^Q_{\mathcal{T}_i}}(f_{\theta'_i}), \tag{2}$$

where η is an outer-loop (or meta) learning rate.



(a) Degree of conflict at each layer (b) Effect of γ on each layer (c) Degree of conflict for each task (d) Generated γ for each task

Fig. 3: **Analysis on degree of conflict and attenuation**. **(a) Degree of conflict** is measured (in radian) and has been observed to vary among layers. For MAML, deeper layers exhibit greater degree of conflict, which aligns with the observation that deeper layers encode more task specific features [7]. After applying L2F to MAML, the degree of conflict has decreased greatly. **(b) The manual attenuation of an initialization** by different levels (*i.e.*, the lower γ is, the stronger is the attenuation) for each layer affects the classification accuracy of a 4-layer CNN on miniImageNet. Deeper layers seem to prefer stronger attenuation. This observation supports the argument that the initialization quality is more compromised in the larger degree of conflict and the compromised part needs to be minimized. **(c) The degree of conflict between each meta-train task and MAML initialization** varies. This observation indicates that the amount of irrelevant prior knowledge differs for each task. **(d) Different attenuation parameters γ are generated by L2F** for each meta-test task, especially for middle-level layers. This differentiation suggests that the degree of conflict, and thus, the preferred amount of attenuation, varies for each task, especially in the middle-level layers.

3.3 Definition of Conflict

While MAML has the merits of simplicity as prior knowledge is only encoded into the common initialization among tasks, we claim that the limitation emanates from the MAML attempting to learn the initialization, which is shared across a distribution of tasks. Although MAML aims to learn a "good" starting point for fast adaptation to new tasks, the shared initialization hinders a fast learning process in actuality. We support our claim by visualizing the sharp optimization landscape during fast adaptation in Figure 2. We hypothesize that the sharp optimization landscape is mainly caused by the disagreement between tasks on the location of a "good" starting point. We call such disagreement as *conflict*.

At each iteration, the feedback from each task $\mathcal{T}_i$ attempts to update the initialization closer to the desired location via the negative loss gradient, which is given by $\boldsymbol{u}_i^Q = -\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^Q}(f_{\theta_i'})$ during the outer-loop optimization. However, as MAML shares the initialization, the initialization is updated via the accumulated gradients over a batch of tasks $\sum_i \boldsymbol{u}_i^Q$, as in Equation 2 (Figure 1). Hence, in the example of a batch of two tasks, *conflict* occurs between tasks $\mathcal{T}_i$ and $\mathcal{T}_j$ when their gradient directions, *i.e.*, the directions of $\boldsymbol{u}_i^Q$ and $\boldsymbol{u}_j^Q$, differ from one another. The more the gradients differ in terms of directions, the more the overall update diverges from $\boldsymbol{u}_i^Q$ and $\boldsymbol{u}_j^Q$, pointing towards the location on the optimization landscape, which is not desirable for both tasks. We refer to this phenomenon as *compromise in the initialization*.

In this study, *the degree of conflict* among tasks is defined as the average angle between $\boldsymbol{u}_i^Q$ and $\sum_i \boldsymbol{u}_i^Q$ and measured as the average arccosine of the dot product of the normalized gradient vectors, $\mathbb{E}_{\mathcal{T}_i \sim p(\mathcal{T})}[\rho_i]$, where ρ_i is $cos^{-1}(\hat{\boldsymbol{u}}_i \cdot \boldsymbol{v})$, $\hat{\boldsymbol{u}}_i$ is $\frac{\boldsymbol{u}_i^Q}{\|\boldsymbol{u}_i^Q\|}$, and $\boldsymbol{v}$ is $\frac{\sum_i \boldsymbol{u}_i^Q}{\|\sum_i \boldsymbol{u}_i^Q\|}$. Figure 3(a) presents the *degree of conflict* measured at each epoch. The figure demonstrates more prominent *conflict* in the deeper layers, which aligns with the previous finding that the deeper layers encode more task-specific features [7].

3.4 Learning to Forget

In the case of *the large degree of conflict* (Figure 1(b)), the initialization is assumed to be more *compromised*, and hence the more difficult it is to quickly learn the new tasks, as illustrated by the sharp loss landscape in Figure 2. The observed sharp loss landscape suggests that the base learner can find some part of the initialization as irrelevant or even detrimental to learning a given task. Thus, in this work, we propose to discard the compromised part of prior knowledge by directly attenuating the initialization parameters θ. Subsequently, one may ask: which part of the initialization is compromised?

To help answer the question, the previous observation on CNN is referred. The first few layers of a CNN encode general knowledge, such as lines, while deeper layers contain more task-specific information, such as a face of an animal [7]. Upon the observation, we hypothesize that deeper layers require more attenuation than shallower layers. We support our hypothesis by varying the amount of attenuation (γ^j) on each layer to observe how much each layer benefits from the scheme (see Figure 3(b)). Corroborating the previous finding [7], deeper layers favor stronger attenuation while shallower layers prefer slight to no attenuation. This difference leads to the second question: how much should the parameters be attenuated on the basis of layers?

One simple answer is to let a meta-learner learn to find an optimal value of attenuation strength for each layer. The answers to the two questions presented above lead to a part of our proposal, *i.e.*, learning layer-wise attenuation by applying a single learnable parameter γ^j on the initialization parameters of each layer θ^j, which is given by

$$\bar{\theta^j} = \gamma^j \theta^j, \tag{3}$$

where j is the layer index of a neural network. The attenuated initialization $\bar{\theta}$ serves as a new starting point for fast adaptation to tasks. While the learned attenuation may reduce the extent of compromise that may exist in the original MAML initialization, the amount of unnecessary

Algorithm 1 Meta-training

Require: Task distribution $p(\mathcal{T})$
Require: Learning rates α, η
Randomly initialize θ, ϕ
Let $\theta = \{\theta^j\}^{j=1...L}$ where j is the layer index and L is the number of layers of a network
while not converged **do**
  Sample a batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for each task $\mathcal{T}_i$ **do**
    Sample examples $(\mathcal{D}^S_{\mathcal{T}_i}, \mathcal{D}^Q_{\mathcal{T}_i})$ from $\mathcal{T}_i$
    Compute $\mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^S_{\mathcal{T}_i}}(f_\theta)$ by evaluating $\mathcal{L}_{\mathcal{T}_i}$ with respect to $\mathcal{D}^S_{\mathcal{T}_i}$ and compute gradients $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^S_{\mathcal{T}_i}}(f_\theta)$
    Take a mean of gradients at each layer and obtain a set of layer-wise mean of gradients: $\{\bar{\boldsymbol{u}}_i^j\}^{j=1...L}$
    Compute attenuation parameter γ for each layer: $\{\gamma_i^j\}^{j=1...L} = g_\phi(\{\bar{\boldsymbol{u}}_i^j\}^{j=1...L})$,
    Attenuate the initialization: $\bar{\theta}_i^j = \gamma_i^j \theta^j$
    Initialize $\theta_i' = \{\bar{\theta}_i^j\}^{j=1...L}$
    for number of inner-loop updates **do**
      Compute $\mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^S_{\mathcal{T}_i}}(f_{\theta_i'})$ by evaluating $\mathcal{L}_{\mathcal{T}_i}$ with respect to $\mathcal{D}^S_{\mathcal{T}_i}$
      Perform gradient descent to compute adapted weights: $\theta_i' = \theta_i' - \alpha \nabla_{\theta_i'} \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^S_{\mathcal{T}_i}}(f_{\theta_i'})$
    end for
    Compute $\mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^Q_{\mathcal{T}_i}}(f_{\theta_i'})$ by evaluating $\mathcal{L}_{\mathcal{T}_i}$ with respect to $\mathcal{D}^Q_{\mathcal{T}_i}$
  end for
  Perform gradient descent to update weights: $(\theta, \phi) \leftarrow (\theta, \phi) - \eta \nabla_{(\theta,\phi)} \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^Q_{\mathcal{T}_i}}(f_{\theta_i'})$
end while

or contradicting information in the initialization can differ among tasks. Figure 3(c) presents the measured degree of conflict for each task, in which the degree of agreement and disagreement with other tasks differs for each task.

For example, there is no consensus between tasks on what the best attenuation is for layer 2, as indicated by different attenuation preferred by each task in Figure 3(d). To resolve the disagreement among tasks, we propose task-adaptive attenuation in addition to layer-wise attenuation. However, this approach poses another question: what kind of task-specific information should be made available for a meta-learner to enable attenuation task-adaptive?

In this work, we turn to gradients $\boldsymbol{u}_i^S = -\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}^S_{\mathcal{T}_i}}(f_\theta)$ of a base learner f_θ for task-specific information. Gradients, which are computed with support examples and used for fast adaptation via the gradient descents, not only hold task-specific information but also encode the quality of the initialization with respect to the given task $\mathcal{T}_i$ from the perspective of optimization. Thus, we propose to compute gradient $\boldsymbol{u}_i^S$ at the initialization and condition a network g_ϕ on it to generate the task-adaptive attenuation. However, the conditioning on the gradients in its raw form entails high complexity. Motivated by the layer-wise behaviours discussed above, g_ϕ instead is conditioned on the layer-wise mean of the gradients, as denoted by

$$\{\gamma_i^j\}^{j=1...L} = g_\phi(\{\bar{\boldsymbol{u}}_i^j\}^{j=1...L}), \tag{4}$$

where γ_i^j is the layer-wise gamma generated for the i-th task at j-th layer, $\bar{\boldsymbol{u}}_i^j$ is the layer-wise mean of gradients $\boldsymbol{u}_i^S$ at the j-th layer, and L is the number of layers of a base learner network f_θ. Here, g_ϕ is a 2-layer MLP network of parameters ϕ, with a sigmoid at the end, to facilitate attenuation.

After the initialization is adapted to each task, the network undergoes fast adaptation (as in Equation 1) and the initialization is updated during training, as in Equation 2. The overall training procedure is summarized in Algorithm 1.

4 Experiments

In this section, we demonstrate the effectiveness and generalizability of our method through extensive experiments by considering various problems, including few-shot classification, cross-domain few-shot classification, regression, reinforcement learning, and visual tracking problem.

4.1 Few-Shot Classification

Few-shot classification is usually formulated as N-way k-shot classification, where each task consists of N number of classes and k number of examples for each class.

4.1.1 Datasets

For the evaluation in the few-shot classification, the most often used datasets are miniImageNet and tieredImageNet, both of which are extracted from ImageNet dataset while taking into account few-shot learning scenarios. Both datasets consist of images with the size of 84 $\times$ 84. The datasets differ in how classes are sampled. miniImageNet randomly samples classes whereas tieredImageNet samples according to the ImageNet class hierarchy to minimize the class similarity between train and test sets, thus rendering the problem even more challenging and realistic.

We further assess our method on two relatively less acknowledged datasets that are derived from CIFAR [42], namely, FC100 (Fewshot-CIFAR100) [18] and CIFAR-FS (CIFAR100 few-shots) [43]. Both datasets aim to create more challenging scenarios by using low resolution images (32$\times$32 compared with 84$\times$84 in miniImageNet and tieredImageNet) from CIFAR100 [42]. These two datasets differ in how they create the train/val/test splits of CIFAR100. While CIFAR-FS follows the procedure used for miniImageNet, FC100 aligns more with the goal of tieredImageNet in that they try to minimize the amount of overlap between splits by performing splitting based on superclasses.

Finally, we perform experiments on a large-scale dataset, Meta-Dataset [3]. Meta-Dataset simulates a more realistic environment by combining several datasets of diverse data distributions and varying the number of classes and examples for each task (*i.e.*, N, k varies for each task).

4.1.2 Results

ImageNet-derived datasets. The results of our proposed approach, other baselines, and existing state-of-the-art approaches on miniImageNet and tieredImageNet with 5-way 1-shot or 5-way 5-shot settings are presented in Table 1. The proposed method improves MAML by a large margin.

TABLE 1: 5-way classification test accuracy on miniImageNet and tieredImageNet

	Backbone	miniImageNet		tieredImageNet	
		1-shot	5-shot	1-shot	5-shot
Matching Network [17]	4 conv	$43.44 \pm 0.77\%$	$55.31 \pm 0.73\%$	–	–
Meta-Learner LSTM [22]	4 conv	$43.56 \pm 0.84\%$	$60.60 \pm 0.71\%$	–	–
MetaNet [19]	5 conv	$49.21 \pm 0.96\%$	–	–	–
LLAMA [38]	4 conv	$49.40 \pm 0.84\%$	–	–	–
Relation Network [16]	4 conv	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$	$54.48 \pm 0.93\%$	$71.32 \pm 0.78\%$
Prototypical Network [15]	4 conv	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$	$53.31 \pm 0.89\%$	$72.69 \pm 0.74\%$
MAML [4]	4 conv	$48.70 \pm 1.75\%$	$63.11 \pm 0.91\%$	$49.06 \pm 0.50\%$	$67.48 \pm 0.47\%$
MAML++ [29]	4 conv	$52.15 \pm 0.26\%$	$68.32 \pm 0.44\%$	–	–
Warp-MAML [35]	4 conv†	$52.3 \pm 0.8\%$	$68.4 \pm 0.6\%$	$57.2 \pm 0.9\%$	$74.1 \pm 0.7\%$
Warp-MAML (reproduced)	4 conv†	$51.70 \pm 0.32\%$	$67.72 \pm 0.24\%$	$55.93 \pm 0.43\%$	$73.50 \pm 0.34\%$
MAML+L2F (Ours)	4 conv	$52.10 \pm 0.50\%$	$69.38 \pm 0.46\%$	$54.40 \pm 0.50\%$	$73.34 \pm 0.44\%$
Warp-MAML+L2F (Ours)	4 conv†	$53.92 \pm 0.47\%$	$69.80 \pm 0.38\%$	$58.86 \pm 0.42\%$	$74.83 \pm 0.37\%$
MetaGAN [39]	ResNet12	$52.71 \pm 0.64\%$	$68.63 \pm 0.67\%$	–	–
SNAIL [40]	ResNet12*	$55.71 \pm 0.99\%$	$68.88 \pm 0.92\%$	–	–
adaResNet [20]	ResNet12	$56.88 \pm 0.62\%$	$71.94 \pm 0.57\%$	–	–
CAML [41]	ResNet12*	$59.23 \pm 0.99\%$	$72.35 \pm 0.71\%$	–	–
TADAM [18]	ResNet12*	$58.5 \pm 0.3\%$	$76.7 \pm 0.3\%$	–	–
MAML	ResNet12	$58.37 \pm 0.50\%$	$69.76 \pm 0.46\%$	$59.15 \pm 0.49\%$	$75.25 \pm 0.43\%$
MAML+L2F (Ours)	ResNet12	$59.71 \pm 0.49\%$	$77.04 \pm 0.42\%$	$64.04 \pm 0.48\%$	$81.13 \pm 0.39\%$
LEO [32]	WRN-28-10*	$61.76 \pm 0.08\%$	$77.59 \pm 0.12\%$	$66.33 \pm 0.05\%$	$81.44 \pm 0.09\%$
LEO (reproduced)	WRN-28-10*	$61.50 \pm 0.17\%$	$77.12 \pm 0.07\%$	$67.02 \pm 0.11\%$	$82.29 \pm 0.16\%$
LEO+L2F (Ours)	WRN-28-10*	$\mathbf{62.12 \pm 0.13}\%$	$\mathbf{78.13 \pm 0.15}\%$	$\mathbf{68.00 \pm 0.11}\%$	$\mathbf{83.02 \pm 0.08}\%$

* a pre-trained network.
† Each layer has 128 channels.

TABLE 2: 5-way classification test accuracy on FC100 and CIFAR-FS

	Backbone	FC100		CIFAR-FS	
		1-shot	5-shot	1-shot	5-shot
MAML *	4 conv	$35.98 \pm 0.48\%$	$51.40 \pm 0.50\%$	$53.91 \pm 0.50\%$	$70.16 \pm 0.46\%$
MAML+L2F (Ours)	4 conv	$\mathbf{39.46 \pm 0.49}\%$	$\mathbf{53.12 \pm 0.50}\%$	$\mathbf{57.28 \pm 0.49}\%$	$\mathbf{73.94 \pm 0.44}\%$
MAML *	ResNet12	$37.92 \pm 0.48\%$	$52.63 \pm 0.50\%$	$64.33 \pm 0.48\%$	$76.38 \pm 0.42\%$
MAML+L2F (Ours)	ResNet12	$\mathbf{41.89 \pm 0.47}\%$	$\mathbf{54.68 \pm 0.50}\%$	$\mathbf{67.48 \pm 0.46}\%$	$\mathbf{82.79 \pm 0.38}\%$

* Our reproduction.

Moreover, L2F maintains a model-agnostic characteristic and achieves a better or comparable accuracy to the state-of-the-art approaches given the same backbone, even without fine-tuning or hyperparameter search. To show the generalization of the contribution, we apply L2F to the state-of-the-art MAML-based systems LEO and Warp-MAML and demonstrate the performance improvement.

CIFAR-derived datasets. Table 2 similarly shows the improvement that has been introduced by L2F over the baseline MAML on FC100 and CIFAR-FS.

Meta-Dataset. Table 3 presents two sub-tables. The top table shows the test accuracy of the models trained on ImageNet (ILSVRC-2012) only, whereas the bottom table includes the test accuracy of the models trained on all datasets (except MSCOCO and Traffic Signs) [3]. The tables also show the classification accuracy and rank, *i.e.*, the average of the datasets, of each model (columns) on the meta-test of each dataset (rows). The baselines are fo-MAML and fo-Proto-MAML. fo-MAML denotes the first-order MAML, whereas fo-Proto-MAML is a fo-MAML variant whose fully-connected (fc) classifier layer is initialized with the Prototypical Network-equivalent weights and bias [3].

Triantafillou *et al.* [3] conducted a hyperparameter search for each model; by contrast, we have not changed the hyper-parameters, as provided by the source code[2]), when applying L2F to each MAML-based model. The effectiveness of the task-and-layer-wise attenuation can be further solidified even in such a large-scale setting given the consistent performance improvement by L2F across the datasets, training settings, and baselines. Moreover, the case of fo-Proto-MAML + L2F outperforming fo-Proto-MAML demonstrates that the proposed attenuation method is complementary to the Prototypical Network-equivalent fc layer initialization in fo-MAML. In [24], MAML initialization was found to be ineffective for feature extractors, which we believe had been due to the MAML initialization compromised by gradient conflicts. With the improved initialization found by L2F, the results suggest that the adaptation of a feature extractor is just as important as that of a classifier.

4.1.3 Ablation Studies

Ablation study results are presented in this section to study the robustness of L2F or investigate the justification of design choice of each module. The ablation study experiments are conducted with a 4-layer CNN in a 5-way 5-shot classification setting on miniImageNet, unless specified otherwise.

Inner-loop update steps. One may argue that the comparisons are not fair because L2F performs one extra adjustment

2. https://github.com/google-research/meta-dataset

TABLE 3: Test accuracy on Meta-Dataset, in which models are trained on ILSVRC-2012 only (top) or all datasets (bottom). The first- and second-best performance are **bolded** and underlined, respectively.

	fo-MAML			fo-Proto-MAML		
	(reported)	(reproduced)	+ L2F	(reported)	(reproduced)	+ L2F
ILSVRC	$45.51 \pm 1.11\%$	$42.82 \pm 1.05\%$	$49.55 \pm 1.19\%$	$49.53 \pm 1.05\%$	$51.03 \pm 1.08\%$	**52.06 ± 1.11%**
Omniglot	$55.55 \pm 1.54\%$	$52.23 \pm 1.51\%$	$69.16 \pm 1.45\%$	$63.37 \pm 1.33\%$	$55.74 \pm 1.41\%$	**70.41 ± 1.43%**
Aircraft	$56.24 \pm 1.11\%$	$46.58 \pm 1.10\%$	$65.70 \pm 1.23\%$	$55.95 \pm 0.99\%$	$51.25 \pm 0.97\%$	**66.52 ± 1.09%**
Birds	$63.61 \pm 1.06\%$	$59.23 \pm 1.15\%$	$62.70 \pm 1.17\%$	$68.66 \pm 0.96\%$	$67.10 \pm 1.05\%$	**69.21 ± 1.04%**
Textures	$68.04 \pm 0.81\%$	$65.19 \pm 0.85\%$	**71.59 ± 0.83%**	$66.49 \pm 0.83\%$	$66.36 \pm 0.83\%$	$71.55 \pm 0.85\%$
Quick Draw	$43.96 \pm 1.29\%$	$46.18 \pm 1.40\%$	$49.46 \pm 1.52\%$	$51.52 \pm 1.00\%$	$56.16 \pm 1.05\%$	**63.01 ± 1.12%**
Fungi	$32.10 \pm 1.10\%$	$30.19 \pm 1.09\%$	$32.42 \pm 1.13\%$	$39.96 \pm 1.14\%$	$38.60 \pm 1.10\%$	**40.71 ± 1.07%**
VGG FLower	$81.74 \pm 0.83\%$	$78.38 \pm 0.90\%$	$87.62 \pm 0.69\%$	$87.15 \pm 0.69\%$	$83.39 \pm 0.78\%$	**87.67 ± 0.78%**
Traffic Signs	$50.93 \pm 1.51\%$	$46.92 \pm 1.53\%$	$50.64 \pm 1.62\%$	$48.83 \pm 1.09\%$	$48.88 \pm 1.05\%$	**64.36 ± 1.22%**
MSCOCO	$35.30 \pm 1.23\%$	$34.49 \pm 1.23\%$	$38.48 \pm 1.33\%$	$43.74 \pm 1.12\%$	$43.73 \pm 1.07\%$	**48.10 ± 1.16%**
Avg. rank	4.3	5.9	3	3.2	3.5	**1.1**

	fo-MAML			fo-Proto-MAML		
	(reported)	(reproduced)	+ L2F	(reported)	(reproduced)	+ L2F
ILSVRC	$37.83 \pm 1.01\%$	$37.55 \pm 1.08\%$	$47.20 \pm 1.19\%$	$46.52 \pm 1.05\%$	$47.03 \pm 1.09\%$	**47.27 ± 1.13%**
Omniglot	$83.92 \pm 0.95\%$	$80.79 \pm 0.98\%$	$83.89 \pm 1.00\%$	$82.69 \pm 0.97\%$	$79.78 \pm 1.07\%$	**84.96 ± 0.92%**
Aircraft	$76.41 \pm 0.69\%$	$72.56 \pm 0.75\%$	$73.85 \pm 1.00\%$	$75.23 \pm 0.76\%$	$68.11 \pm 0.88\%$	**78.05 ± 0.83%**
Birds	$62.43 \pm 1.08\%$	$61.61 \pm 0.93\%$	$65.86 \pm 1.06\%$	$69.88 \pm 1.02\%$	$67.88 \pm 1.01\%$	**71.80 ± 0.93%**
Textures	$64.14 \pm 0.83\%$	$64.03 \pm 0.81\%$	**70.25 ± 0.97%**	$68.25 \pm 0.81\%$	$65.09 \pm 0.83\%$	$69.90 \pm 0.80\%$
Quick Draw	$59.73 \pm 1.10\%$	$59.48 \pm 1.07\%$	$61.66 \pm 1.28\%$	$66.84 \pm 0.94\%$	$68.20 \pm 0.87\%$	**70.71 ± 0.89%**
Fungi	$33.54 \pm 1.11\%$	$31.35 \pm 1.10\%$	$35.65 \pm 1.16\%$	$41.99 \pm 1.17\%$	$41.90 \pm 1.12\%$	**43.85 ± 1.16%**
VGG FLower	$79.94 \pm 0.84\%$	$79.67 \pm 0.81\%$	$87.93 \pm 0.66\%$	**88.72 ± 0.67%**	$87.12 \pm 0.74\%$	$88.36 \pm 0.63\%$
Traffic Signs	$42.91 \pm 1.31\%$	$41.48 \pm 1.25\%$	$52.34 \pm 1.35\%$	$52.42 \pm 1.08\%$	$50.44 \pm 1.05\%$	**61.58 ± 1.23%**
MSCOCO	$29.37 \pm 1.08\%$	$31.74 \pm 1.14\%$	$38.22 \pm 1.22\%$	$41.74 \pm 1.13\%$	$42.59 \pm 1.05\%$	**44.66 ± 1.08%**
Avg. rank	4.4	5.6	3.6	2.6	3.6	**1.2**

TABLE 4: Ablation studies on inner-loop update steps on 5-way 5-shot miniImageNet classification.

Inner-loop update steps	MAML	MAML+L2F(Ours)
1	$56.93 \pm 0.32\%$	$68.16 \pm 0.47\%$
2	$55.63 \pm 0.50\%$	$66.85 \pm 0.49\%$
3	$58.79 \pm 0.49\%$	$68.61 \pm 0.46\%$
4	$62.72 \pm 0.45\%$	$68.66 \pm 0.43\%$
5	$63.94 \pm 0.41\%$	$69.38 \pm 0.46\%$
6	$64.54 \pm 0.46\%$	–

to initialization parameters before inner-loop updates. Table 4 is presented to address this concern. The table shows the ablation studies of a number of inner-loop updates for the proposed model and the baseline. The results indicate that the performance gain is not due to the extra number of adjustments of the parameters. Rather, the benefits come from *forgetting* the unnecessary information, helping the learner to quickly adapt to new tasks.

Attenuation scope. One may ask: is layer-wise attenuation the best way to apply attenuation? To answer this question, we analyze different scopes of attenuation: a single attenuation parameter for the whole network, an individual attenuation parameter for each layer, each filter (channel), and each weight of the network. To focus on investigating which scope of attenuation is the most beneficial, we remove the task-adaptive part and make the attenuation parameters learnable, and thus task-agnostic (with values initialized as 1), rather than generated by the network g_ϕ.

Table 5 summarizes the ablation study results of the attenuation scope. The layer-wise attenuation provides the highest performance gain. As discussed in Appendix, attenuation can also be considered as meta-level regularization that controls the amount of prior knowledge (learned by initialization) that will be used for fast adaptation. Controlling the values of more parameters with a single value amounts to regularization of the greater extent (*i.e.*, the parameter-wise attenuation gives the least amount of regularization while the network-wise attenuation the largest). Within these two extreme ends, the layer-wise attenuation is empirically shown in Table 5 to strike a balance and provide the highest performance boost, which is consistent with the observation that the level of feature hierarchies (and thus the extent of task-specific information) varies by the depth of layers [7].

TABLE 5: Ablation studies on attenuation scope. Except for MAML+L2F, all models learn task-agnostic attenuation parameters to illustrate the effect of attenuation scope alone, without task-conditioning.

Attenuation Scope	Accuracy
None (MAML, our reproduction)	$63.94 \pm 0.48\%$
parameter-wise	$64.70 \pm 0.43\%$
filter (channel)-wise	$65.35 \pm 0.48\%$
layer-wise	**68.49 ± 0.41%**
network-wise	$67.84 \pm 0.46\%$
MAML+L2F (Ours)	**69.38 ± 0.46%**

Effect of task-conditioning. Table 5 reports the lower performance of the task-agnostic layer-wise attenuation model, compared with our full task-adaptive model (MAML+L2F). The only difference between the layer-wise attenuation model and our model is that the former lacks the task-conditioning.

TABLE 6: Ablation studies on the types of representation for task embedding

Task Embedding	Accuracy
Features (class prototype)	$68.73 \pm 0.46\%$
Gradients (Ours, MAML+L2F)	$\mathbf{69.48 \pm 0.46}\%$

TABLE 7: Ablation studies on task-adaptive transformation depicting the effectiveness of attenuation.

Model	Description	Accuracy
1	MAML (our reproduction)	$63.94 \pm 0.48\%$
2	MAML + task-adaptive non-sigmoided γ_i^j, δ_i^j	$66.22 \pm 0.47\%$
3	MAML + task-adaptive non-sigmoided γ_i^j	$67.56 \pm 0.47\%$
Ours	MAML + L2F (task-adaptive sigmoided γ_i^j)	$\mathbf{69.38 \pm 0.46}\%$

One can observe that the highest performance gain in our method emanates from the attenuation, alluding to the importance of attenuation. Regardless, the task-conditioning can also improve the performance.

Representation of task embedding. In justifying the use of gradients for task representation, we compare it with alternative representation, which is the mean of class prototypes from the pre-trained prototypical network [15], similar to TADAM [18]. Table 6 demonstrates that our method with gradients as task representation performs slightly better than the one with the mean of class prototypes. This finding justifies choosing gradients for task representation, especially because gradients provide higher performance and are model-agnostic and flexible, whereas class prototypes are inapplicable across different domains.

Effect of attenuation. To further analyze the effectiveness of the attenuation of L2F, we apply various alternative task-adaptive transformations to MAML. The results are presented in Table 7. We start with the simple superset of the attenuation γ, which is scaling without sigmoid (Model 3) such that γ_i is no longer restricted to be between 0 and 1 and hence does not facilitate attenuation. We also explore a more flexible option via affine transformation (Model 2), in which the network g_ϕ generates two sets of parameters γ_i, δ_i without sigmoid, which will modulate f_θ via $\gamma_i^j \theta^j + \delta_i^j$.

Table 7 illustrates that MAML can gain performance boosts throughout the different types of task-adaptive transformation, suggesting the benefits of the task-conditioning. It is reasonable to expect that more flexible transformations (Model 2 and 3) allow for tasks to bring the initialization to a more appropriate location for fast adaptation. Interestingly, the accuracy decreases as more flexibility is given to the transformation of the initialization. This seeming contradiction underlines the necessity of the attenuation (sigmoided γ_i^j in our model) of the initialization to *forget* the compromised part of the prior knowledge encoded in the initialization rather than simply relying on the naïve transformation.

Notably, MAML with task-agnostic layer- or network-wise attenuation (see Table 5) performs better than the other task-adaptive transformations in Table 7. This finding suggests that *forgetting* the compromised initialization is more important than making it task-adaptive.

4.2 Cross-Domain Few-Shot Classification

To examine the capability of meta-learning algorithms under more realistic few-shot learning scenarios, Chen *et al.* [44] introduced the cross-domain few-shot classification setting, in which meta-learners could be trained on the miniImageNet meta-train set but tested on the CUB (CUB-200-2011) meta-test set [45]. This type of problem setting can minimize the overlap between the data distributions of the meta-train and meta-test, allowing for the improved evaluation of the ability of meta-learners to learn new tasks. The philosophy of cross-domain few-shot classification is similar to that of Meta-Dataset, in which one of the experiments involves training the models only on ImageNet but evaluated across diverse datasets, as demonstrated by the top table in Table 3.

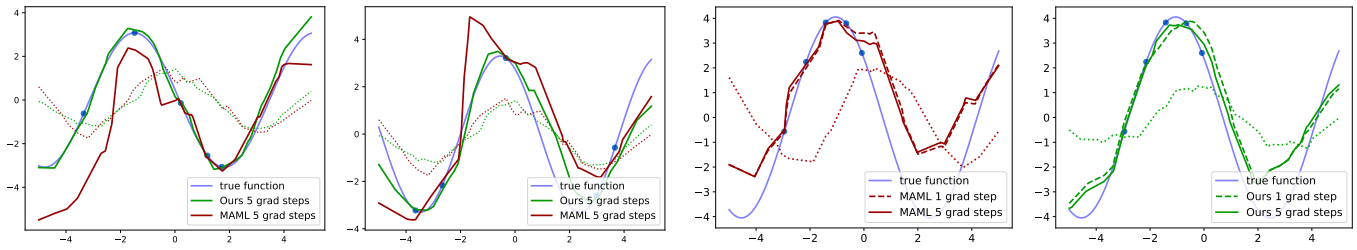
TABLE 8: Test accuracy on 5-way 5-shot cross-domain few-shot classification. Each model is trained on the miniImageNet meta-train set and tested on the CUB meta-test set.

	Backbone	**miniImageNet → CUB**
MAML	4 conv	$52.70 \pm 0.32\%$
MAML + L2F	4 conv	$\mathbf{60.89 \pm 0.22}\%$
MAML	ResNet12	$54.06 \pm 0.49\%$
MAML + L2F	ResNet12	$\mathbf{63.64 \pm 0.42}\%$

Table 8 outlines the results. L2F substantially improves the performance of the baseline MAML; in fact, the performance gain is larger than that when the model has been tested on the same dataset (Table 1). The finding depicting L2F being more effective when the data distribution gap is large between meta-train and meta-test suggests that L2F imbues a meta-learner with better capability of handling a new task owing to its task-adaptive control over prior knowledge in the form of initialization.

4.3 Regression

We investigate the generalizability of the proposed method across diverse problems, starting with an evaluation of the performance in the k-shot regression. In k-shot regression, the objective is to fit a function, given the k sampled points of the function. Following the general settings in [4], [28], the target function is set as a sinusoid with varying amplitudes and phases between tasks. The sampling ranges of the amplitude, frequency, and phase define the task distribution, in which the ranges are set to be the same for both meta-training and meta-testing (evaluation). The qualitative results are visualized in Figure 4(a). The results demonstrate that MAML+L2F not only converges faster but also fits to the target functions more accurately than MAML. The quantitative result presented in Table 9 also shows the significant performance improvement introduced by L2F to MAML. Furthermore, the proposed method outperforms the improved versions of MAML that can work in diverse problem domains, namely MMAML [31] and MAML++ [32]. MAML++ improves the inner-loop optimization by providing meta-update supervision at each inner-loop step and meta-learning layer-and-step-wise inner-loop learning rate. On other hand, MMAML transforms the initialization via the affine transformation according to each task. The finding on MAML+L2F outperforming both methods further signifies the importance of applying an attenuation on the initialization rather than by simply performing a task-adaptive transformation of the initialization or improving the inner-loop optimization.



(a) Fully-overlapped task distributions (b) Disjoint task distributions

Fig. 4: **5-shot regression.** The task distributions for training and evaluation are either (a) identical or (b) disjoint, with no overlap. In both cases, MAML+L2F (our model) is more fitted to the true function. The small dotted lines represent the regression results with initial weights before inner-loop optimization but excluded from the legends for clear visualization).

TABLE 9: MSE averaged over 100 sampled points with 95%confidence intervals on 5-shot regression. Our method consistently outperforms other methods across all steps.

Models	1 step	2 steps	5 steps
MAML [4]	1.2247	1.0268	0.8995
MMAML [31]	1.1010	0.9291	0.8615
MAML++ [29]	1.2028	0.9268	0.7547
Warp-MAML [35]	1.0842	0.7742	0.6145
MAML+L2F (Ours)	1.0537	0.8426	0.7096
Warp-MAML+L2F (Ours)	**0.9379**	**0.7101**	**0.4829**

TABLE 10: Average reward reported for 2D navigation task. L2F consistently outperforms other methods across all steps.

Models	1 step	2 steps	3 steps
MAML [4]	-32.626	-25.746	-20.734
MMAML [31]	-25.785	-23.705	-19.747
MAML++ [29]	-36.281	-27.264	-18.620
Warp-MAML [35]	-26.693	-22.889	-17.668
MAML+L2F (Ours)	-24.230	-19.585	-16.517
Warp-MAML+L2F (Ours)	**-21.356**	**-17.176**	**-15.395**

With the similar motivation as cross-domain few-shot classification, this study also presents a modified setting, in which the amplitude, frequency, and phase are sampled from the non-overlapped task distributions for training and evaluation to minimize the overlap between the meta-train and meta-test data distributions. The minimized overlap between meta-training and meta-testing data distributions effectively increases the degree of conflict between new tasks and prior knowledge, allowing for the enhanced assessment of the capability of meta-learning algorithms to learn new tasks. In Figure 4(b), MAML+L2F exhibits higher accuracy and thus claims better generalization, further underlining the effectiveness of the attenuation in handling new tasks, even when they are sampled from the task distribution disjoint from the one used during meta-training.

4.4 Reinforcement Learning

To further validate the generalizability of L2F, we evaluate its performance in reinforcement learning, specifically 2D navigation and locomotion tasks [46], as in [4]. The task description is briefly outlined in each corresponding section. Figure 5 presents the consistent improvement of L2F over MAML across the different experiments, solidifying the generalizability and effectiveness of our proposed method.

4.4.1 2D Navigation

In a 2D navigation task, an agent is devised to move from the given starting point to the given ending point in 2D space, and the reward is defined as the negative of the squared distance to the destination point. Each task is defined as the set of the starting point coordinates and the ending point coordinates. We follow the experiment procedure in [4], with a fixed starting point ($[0, 0]$), and we only vary the ending point coordinates between tasks ($[-0.5, 0.5] \times [-0.5, 0.5]$).

Figure 5(a) presents the much more precise navigation by MAML+L2F, both quantitatively (left) and qualitatively (right). The trends solidify the severity of the conflicts between tasks and the importance of attenuation. Additional quantitative comparisons with other MAML variants, namely, MMAML [31] and MAML++ [29], are presented in Table 10. The results from Table 10 share the same tendency as those from Table 9 in that the proposed method consistently outperforms other MAML-based algorithms across gradient steps. Thus, L2F has consistently validated the importance of attenuation on the initialization, outperforming other MAML variants across different problem domains.

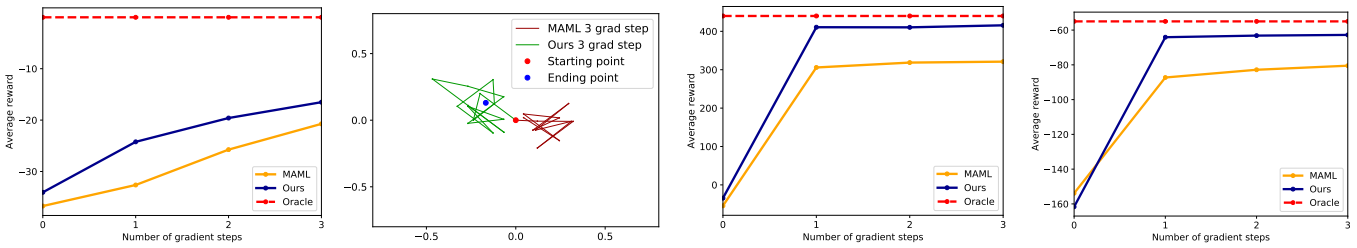
Figure 6 presents the few other qualitative results. Both training and evaluation tasks in Figure 6(a) are sampled from the same distribution, in which the starting point is fixed to $[0, 0]$ and the ending point is sampled in the range of $[-0.5, 0.5] \times [-0.5, 0.5]$. Then, in Figure 6(b), the training tasks are sampled from the same distribution as that in Figure 6(a), whereas the evaluation tasks are sampled from different distribution, *i.e.*, both the starting and ending point coordinates are sampled in the range of $[-0.5, 0.5] \times [-0.5, 0.5]$. This approach is designed to increase the disparities between the new unseen tasks and prior knowledge. In both scenarios, the proposed method can quickly reach the ending point, whereas MAML often fails to reach the destination.

4.4.2 MuJoCo

Locomotion experiments are further conducted using the MuJoCo simulator [47] to evaluate the meta-learners on a much more complex reinforcement-learning environment. The two sets of tasks are as follows: a robot (half-cheetah) is devised to move in a certain direction in the first set and move with a certain velocity in the second set. In both experiments, the proposed method outperforms MAML by a large margin, as shown in Figure 5 (b), (c).

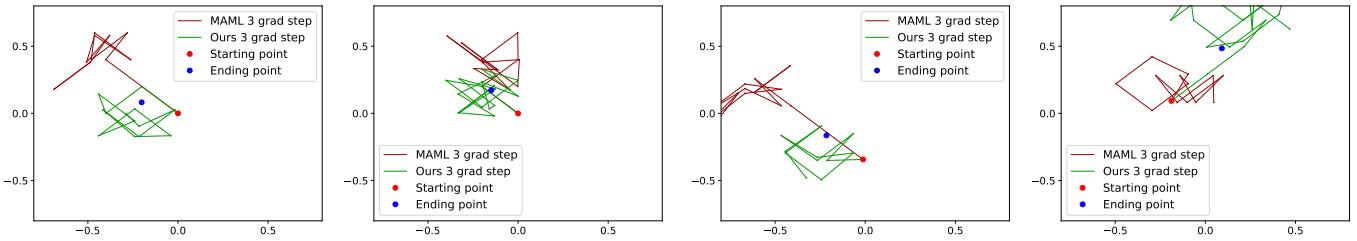
4.4.3 RLBench

To further examine the capability of MAML and L2F to quickly adapt to various tasks in the context of reinforce-



(a) 2D navigation (b) Half-cheetah (direction) (c) Half-cheetah (velocity)

Fig. 5: **Reinforcement learning results** in three different environments. MAML+L2F outperforms MAML.



(a) Start $= [0, 0]$, End $\in [-0.5, 0.5] \times [-0.5, 0.5]$ (b) Start, End $\in [-0.5, 0.5] \times [-0.5, 0.5]$

Fig. 6: **2D navigation reinforcement learning task.** The task distribution during evaluation is (a) the same as or (b) different from training. In both cases, MAML+L2F reaches the ending point, whereas MAML often fails to reach the destination.

TABLE 11: Average return on FS10-V1 few-shot benchmark.

	MAML	MAML+L2F
Average return	0.2963	**0.4580**

ment learning, we evaluate them on a recently introduced large-scale robot learning benchmark, named RLBench [48]. RLBench introduces various tasks and challenges, including few-shot challenges. Similar to cross-domain few-shot classification [44] and Meta-Dataset [3], the RLBench few-shot benchmark simulates the challenging scenarios by sampling train and test tasks from two different task distributions. In particular, we evaluate MAML and L2F on a FS10-V1 few-shot benchmark[3] that uses 10 major tasks (*ReachTarget*, *CloseBox*, *ClosMicrowave*, *PlugChargerInPowerSupply*, *ToiletSeatDown*, *TakeUmbrellaOutofUmbrellaStand*, *SlideCabinetOpen*, *CloseFridge*, and *PickAndLift*) for meta-training and 5 major tasks for meta-test (*OpenBox*, *OpenMicrowave*, *UnplugCharger*, *ToiletSeatUp*, and *OpenFridge*). Another challenging aspect is that reward is sparse (1 only when the goal is achieved).

Table 11 presents the average return, where the evaluation is performed with the model with the highest average training return. The significant performance boost shown in Table 11 solidifies the effectiveness of L2F in handling gradient conflicts that are more likely to occur in the environments, such as RLBench, that have diverse tasks.

4.5 Visual Object Tracking

4.5.1 Online adaptation and meta-learning

Visual tracking challenges, such as similar-looking distractors and dynamic appearance changes of the target, require trackers to be adaptable and robust as a means of successfully tracking any given target without drifts. One approach to achieve the adaptability and robustness is online adaptation, which updates the models of the target at test time, where the tracking models are often formulated as discriminative correlation filters or binary classifiers (the target and background as two classes) to discriminate the target from the background and distractors. Such an online adaptation approach for visual tracking shares some of its philosophy with meta-learning. Both of them aim to tackle problems involving tasks that are not known *a priori*, and they also address the challenge through online adaptation at a given test time. Park and Berg [49] proposed to apply meta-learning to two of the state-of-the-art online-adaptation-based trackers, CREST [50] and MDNet [51], to improve their tracking performance. In particular, they used MAML as the online adaptation module and named the variant of each tracker as MetaCREST and MetaSDNet. Meta-Tracker utilizes MAML to find a good set of initialization weights for the tracking models, from which the models fine-tune to each tracking task with the given bounding box at an initial frame and generalize well across future frames.

As the MAML can be used as a meta-learner for meta-trackers, simple extensions to MAML can also be easily applied to Meta-Tracker. Tseng *et al.* [52] applied their improved version of MAML named DropGrad to each Meta-Tracker (MetaCREST and MetaSDNet). In particular, they improved MAML by regularizing the inner-loop optimization through inner-loop gradient dropout. Similar to DropGrad, L2F is a simple extension to MAML and can be easily applied to Meta-Trackers. Furthermore, L2F can be applied in conjunction with DropGrad because they focus on two complementary parts of MAML, namely initialization and inner-loop optimization.

4.5.2 Datasets

Following the settings in [49], the models are trained with a subset of 718 video sequences from a large-scale ImageNet Video detection dataset [53] in addition to the 58 sequences from the visual object tracking benchmark datasets obtained

3. https://github.com/stepjam/RLBench

TABLE 12: Precision and success rate measured over 100 sequences in the OTB2015 dataset by using OPE. The first- and second-best performances are **bolded** and underlined.

Model	Precision	Success rate
MetaCREST [49]	0.7994	0.6029
MetaCREST + L2F	0.8355	0.6247
MetaCREST + DropGrad [52]	0.8172	0.6145
MetaCREST + DropGrad + L2F	**0.8506**	**0.6261**
MetaSDNet [49]	0.8673	0.6434
MetaSDNet + L2F	0.8835	0.6533
MetaSDNet + DropGrad [52]	0.8746	0.6520
MetaSDNet + DropGrad + L2F	**0.8948**	**0.6590**

from VOT2013 [54], VOT2014 [55], and VOT2015 [56], excluding the video sequences from OTB2015 [57].

4.5.3 Results

The trackers are evaluated via the one-pass evaluation (OPE) protocol, which means that the trackers do not restart after the initialization at the first frame, even at failures [57]. Table 12 lists the results after the application of L2F to either Meta-Tracker or Meta-Tracker+DropGrad. The consistent improvement by L2F further reinforces the generalizability and effectiveness of the task-adaptive attenuation of initialization for learning new tasks. Meta-Trackers and their variants are trained using the source code provided by the authors[4]. All trackers are trained with the default set of hyperparameters.

4.6 Loss Landscape

The effectiveness of L2F is further validated by illustrating the smoother loss landscape after applying L2F to MAML, as previously shown in Figure 2. At the beginning of the training, L2F appears to struggle more with the sharper loss landscape than MAML. The observation may seem contradictory at first, but it further validates our argument on *conflicts* between tasks. At the initial training stage, the MAML initialization is not trained enough and thus does yet not have the sufficient amount of prior knowledge of the task distribution. As the training proceeds, the initialization encodes additional information about task distribution and begins to frequently encounter conflicts between tasks. As for L2F, the attenuator g_ϕ initially does not have enough knowledge about the task distribution and thus generates inaccurate attenuation parameters γ_i, deteriorating the initialization. However as the training continues, the attenuator encodes additional information about the task distribution, thus generating more appropriate attenuation γ_i that corresponds well to the tasks and layers. The generated γ_i allows for the learner to selectively *forget* the irrelevant part of prior knowledge as a means of facilitating fast adaptation, as illustrated by the increasing stability and smoothness of the landscape in Figure 2.

5 Conclusion

This work argues that forcibly sharing initialization across tasks in MAML induces conflicts among tasks, thereby misguiding the initialization to the compromised location. The observed sharp loss landscape supports our claim that such a compromise renders the MAML initialization a "bad" starting position for generalization after fast adaptation with few examples. To resolve the discrepancy, we propose a meta-learner that learns to *forget* the irrelevant information that may hinder fast adaptation. In particular, we propose a task-and-layer-wise attenuation named L2F to facilitate *selective forgetting*. The extensive experiments across diverse domains (*e.g.*, visual tracking) validate the simplicity, effectiveness, and generalizability of L2F.

4. https://github.com/silverbottlep/meta_trackers

Acknowledgments

This work was supported in part by IITP grant funded by the Korea government [No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)], and in part by AIRS Company in Hyundai Motor and Kia through HMC/KIA-SNU AI Consortium Fund.

References

[1] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR Workshop*, 2014.
[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
[3] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, and H. Larochelle, "Meta-dataset: A dataset of datasets for learning to learn from few examples," in *ICLR*, 2020.
[4] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.
[5] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization?" in *NeurIPS*, 2018.
[6] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *NeurIPS*, 2018.
[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.
[8] S. Baik, S. Hong, and K. M. Lee, "Learning to forget for meta-learning," in *CVPR*, 2020.
[9] S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei, "On the optimization of a synaptic learning rule," *Optimality in Artificial and Biological Neural Networks*, 1997.
[10] S. Hochreiter, A. Younger, and P. Conwell, "Learning to learn using gradient descent," *Artificial Neural Networks, ICANN 2001*, 2001.
[11] J. Schmidhuber, "Evolutionary principles in self-referential learning," *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1987.
[12] ——, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Computation*, 1992.
[13] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.
[14] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, 2015.
[15] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017.
[16] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018.
[17] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *NIPS*, 2016.
[18] B. N. Oreshkin, P. Rodriguez, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *NeurIPS*, 2018.
[19] T. Munkhdalai and H. Yu, "Meta networks," in *ICML*, 2017.
[20] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler, "Rapid adaptation with conditionally shifted neurons," in *ICML*, 2018.
[21] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *ICML*, 2016.
[22] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017.
[23] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv:1803.02999*, 2018.
[24] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of maml," in *ICLR*, 2020.

[25] M. A. Jamal and G.-J. Qi, "Task agnostic meta-learning for few-shot learning," in *CVPR*, 2019.
[26] M. Yin, G. Tucker, M. Zhou, S. Levine, and C. Finn, "Meta-learning without memorization," in *ICLR*, 2020.
[27] L. M. Zintgraf, K. Shiarlis, V. Kurin, K. Hofmann, and S. Whiteson, "Fast context adaptation via meta-learning," in *ICML*, 2019.
[28] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few shot learning," *arXiv:1707.09835*, 2017.
[29] A. Antoniou, H. Edwards, and A. Storkey, "How to train your maml," in *ICLR*, 2019.
[30] Y. Lee and S. Choi, "Gradient-based meta-learning with learned layerwise metric and subspace," in *ICML*, 2018.
[31] R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim, "Multimodal model-agnostic meta-learning via task-aware modulation," in *NeurIPS*, 2019.
[32] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," in *ICLR*, 2019.
[33] H. Yao, Y. Wei, J. Huang, and Z. Li, "Hierarchically structured meta-learning," in *ICML*, 2019.
[34] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *CVPR*, 2019.
[35] S. Flennerhag, A. A. Rusu, R. Pascanu, F. Visin, H. Yin, and R. Hadsell, "Meta-learning with warped gradient descent," in *ICLR*, 2020.
[36] Z. Chen, J. Ngiam, Y. Huang, T. Luong, H. Kretzschmar, Y. Chai, and D. Anguelov, "Just pick a sign: Optimizing deep multitask models with gradient sign dropout," in *NeurIPS*, 2020.
[37] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *NeurIPS*, 2020.
[38] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," in *ICLR*, 2018.
[39] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *NeurIPS*, 2018.
[40] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *ICLR*, 2018.
[41] X. Jiang, M. Havaei, F. Varno, G. Chartrand, N. Chapados, and S. Matwin, "Learning to learn with conditional class dependencies," in *ICLR*, 2019.
[42] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.
[43] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *ICLR*, 2019.
[44] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *ICLR*, 2019.
[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Caltech, Tech. Rep., 2011.
[46] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *ICML*, 2016.
[47] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IROS*, 2012.
[48] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, 2020.
[49] E. Park and A. C. Berg, "Meta-tracker: Fast and robust online adaptation for visual object trackers," in *ECCV*, 2018.
[50] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. H. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *ICCV*, 2017.
[51] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016.
[52] H.-Y. Tseng, Y.-W. Chen, Y.-H. Tsai, S. Liu, Y.-Y. Lin, and M.-H. Yang, "Regularizing meta-learning via gradient dropout," *arXiv:2004.05859*, 2020.
[53] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, 2015.
[54] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Čehovin Zajc, G. Nebehay, G. Fernandez, and T. V. et al., "The visual object tracking vot2013 challenge results," in *ICCV Workshop*, 2013.
[55] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin Zajc, G. Nebehay, T. Vojir, G. Fernandez, and A. L. et al., "The visual object tracking vot2014 challenge results," in *ECCV Workshop*, 2014.
[56] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin Zajc, G. Fernandez, T. Vojir, G. Häger, and G. N. et al., "The visual object tracking vot2015 challenge results," in *ICCV Workshop*, 2015.
[57] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

Sungyong Baik received the BASc degree in engineering science with a major in electrical and computer engineering from University of Toronto, Toronto, Canada, in 2015. He is currently working towards the PhD degree in electrical and computer engineering from Seoul National University. He was a research intern at Facebook Reality Labs in 2019. His research interests include few-shot learning, meta-learning, and its applications. He is a student member of the IEEE.

Junghoon Oh received BS degree in electrical and computer engineering from Seoul National University (SNU), Seoul, Korea, in 2020. He is currently working towards the MS degree in electrical and computer engineering from Seoul National University. His research interests include meta-learning and network pruning. He is a student member of the IEEE.

Seokil Hong received BS degree in electrical and computer engineering from Seoul National University (SNU), Seoul, Korea, in 2018. He is currently working towards the PhD degree in electrical and computer engineering from Seoul National University. His research interests include computer vision, machine learning, and deep learning. He is a student member of the IEEE.

Kyoung Mu Lee received the BS and MS degrees in control and instrumentation engineering from Seoul National University (SNU), Seoul, Korea, in 1984 and 1986, respectively, and the PhD degree in electrical engineering from the University of Southern California, in 1993. He is currently with the Department of ECE, Seoul National University as a professor. He has received several awards, in particular, the medal of merit and the Scientist of Engineers of the month award from the Korean Government in 2020 and 2018, respectively, the Most Influential Paper over the Decade Award by the IAPR Machine Vision Application in 2009, the ACCV Honorable Mention Award in 2007, the Okawa Foundation Research Grant Award in 2006, the Distinguished Professor Award from the college of Engineering of SNU in 2009, and both the Outstanding Research Award and the Shinyang Engineering Academy Award from the College of Engineering of SNU in 2010. He has served as an Associate Editor in Chief (AEIC) and an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), an Associate Editor of the Machine Vision Application (MVA) Journal and the IPSJ Transactions on Computer Vision and Applications (CVA), the IEEE Signal Processing Letters (SPL), and an Area Editor of the Computer Vision and Image Understanding (CVIU). He also has served as a general chair of ICCV2019, ACMMM2018, and ACCV2018, a program chair of ACCV2012, a track chair of ICPR2020 and ICPR2012, and an area char of CVPR, ICCV and ECCV many times. He was a distinguished lecturer of the Asia-Pacific Signal and Information Processing Association (APSIPA) for 2012-2013. More information can be found on his homepage http://cv.snu.ac.kr/kmlee.