Learning to Forget for Meta-Learning

Sungyong Baik Seokil Hong Kyoung Mu Lee
ASRI, Department of ECE, Seoul National University
{dsybaik, hongceo96, kyoungmu}@snu.ac.kr

Abstract

Few-shot learning is a challenging problem where the goal is to achieve generalization from only few examples. Model-agnostic meta-learning (MAML) tackles the problem by formulating prior knowledge as a common initialization across tasks, which is then used to 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 the task adaptation. Further, we observe that the degree of conflict differs among not only tasks but also layers of a neural network. Thus, we propose task-and-layer-wise attenuation on the compromised initialization to reduce its influence. As the attenuation dynamically controls (or selectively forgets) the influence of prior knowledge for a given task and each layer, we name our method as L2F (Learn to Forget). The experimental results demonstrate that the proposed method provides faster adaptation and greatly improves the performance. Furthermore, L2F can be easily applied and improve other state-of-the-art MAML-based frameworks, illustrating its simplicity and generalizability.

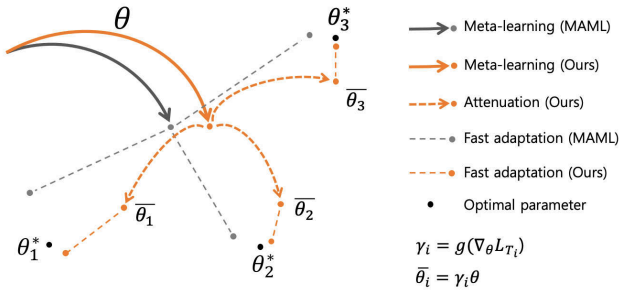


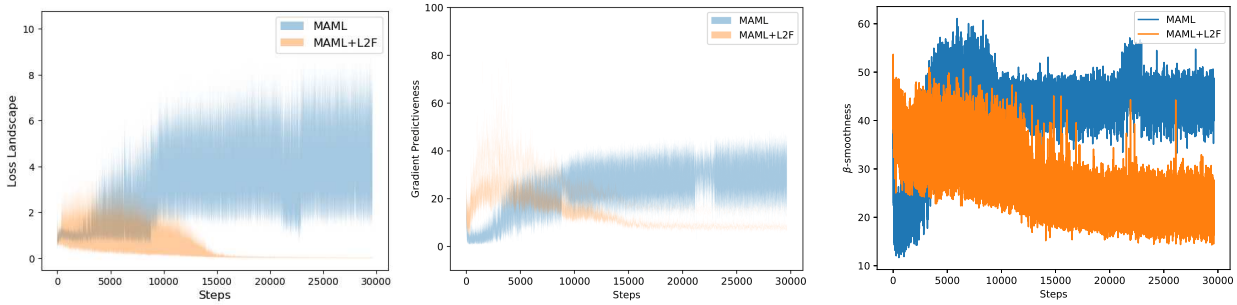
Figure 1: When there is a large degree of *conflict*, the updated initialization ends up in the location neither of tasks desires. Such undesired (hence compromised) initialization location can make learning difficult during fast adaptation to each task. Our method makes the fast adaptation easier by minimizing the influence of the compromised initialization for each task, through attenuation parameter γ generated by the task-conditioned network g. This makes the optimization landscape smoother and hence helps achieve better generalization to unseen examples.

1. Introduction

Recent deep learning models demonstrate outstanding performance in various fields; however, they require supervised learning with a tremendous amount of labeled data. On the other hand, humans are able to learn concepts from only few examples. Considering the cost of data annotation, the capability of humans to learn from few examples is desirable.

When there are concerns for overfitting in few-data regime, data augmentation and regularization techniques are often used. Another commonly used technique is to fine-tune a network pre-trained on large labelled data from another dataset or task [19, 26]. Fine-tuning often does provide adaptation without overfitting even in few-data regime, however at the cost of computation due to many update iterations [31]. In contrast, meta-learning tackles the problem systematically via two stages of learners: a meta-learner learns common knowledge across a distribution of tasks, which is then used for a learner to quickly learn task-specific knowledge with few examples. A popular instance is the model-agnostic meta-learning (MAML) [5], where a meta-learner is formulated such that it learns a common initialization that encodes the common knowledge across tasks.

The assumption of the existence of a task distribution may justify MAML for seeking a common initialization among tasks. But, there still exists variations among tasks, some of which may lead to the disagreement among tasks on the location of the initialization. We call such disagreement *conflict* and formally define it in this paper. Some of prior knowledge encoded in such compromised initialization is useful for one task but may be irrelevant or even detrimental for another. Consequently, a learner struggles to learn new concepts quickly with the prior knowledge that conflicts with information from new examples, as illustrated in Fig-



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

Figure 2: Visualization of optimization landscape: In [23], they analyze the stability and smoothness of the optimization landscape by measuring Lipschitzness and the "effective" β-smoothness of loss. We use these measurements to analyze learning dynamics for both MAML and our proposed method during training on 5-way 5-shot miniImageNet classification tasks, i.e. investigating fast-adaptation (or inner-loop) optimization. At each inner-loop update step, we measure 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 training iterations. The thinner shade in plots (a) and (b) and the lower the values in plot (c) indicate the smoother loss landscape and thus less training difficulty [23].

ure 1. Such learning difficulty can manifest as the sharp loss landscape and thereby poor generalization to new examples [11, 23]. Motivated by our hypothesis, we analyze and indeed observe the sharp landscape during fast adaptation to new examples (as shown in Figure 2) and suggest that the learned initialization by MAML is a "bad" location.

One solution for a meta-learner would be to simply *forget* the part of the initialization that hinders adaptation to the task, minimizing its influence. This raises two questions: Where do these *conflicts* occur? To what extent? We hypothesize that the degree of conflict varies among layers of a neural network, especially CNN, since deeper layers learn more task-specific knowledge or class-specific knowledge in classification [34]. To test the hypothesis, we measure *conflict* at each layer and observe that *conflict* is indeed more severe at deeper layers, as shown in Figure 3(a). We also observe that the amount of agreement between the learned initialization and the initialization desired by a given task differs for each task in Figure 3(c). Thus, we argue that *conflicts* occur at two levels: task and layer.

Motivated by the observation, we propose to learn selective *forgetting* by applying a task-and-layer-wise *attenuation* on MAML initialization, controlling the influence of prior knowledge for each task and layer. For each task, we argue that initialization weights and its gradients (obtained from support examples of task), together, encode information about optimization specific to a task, and thus 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 L2F (Learn to Forget), indeed improves the quality of the initialization (illustrated by a smoother loss landscape in Figure 2) and consistent performance improvement across different domains, managing to maintain the simplicity and generalizability of MAML.

2. Related Work

Meta-learning aims to learn across-task prior knowledge to achieve fast adaptation to specific tasks [2, 7, 24, 25, 29]. Recent meta-learning systems can be broadly classified into three categories: metric-based, network-based, and optimization-based. The goal of metric-based system is to learn relationship between query and support examples by learning an embedding space, where similar classes are closer and different classes are further apart [9, 27, 28, 32]. Network-based approaches encode fast adaptation into network architecture, for example, by generating input-conditioned weights [14, 17] or employing an external memory [15, 22]. On the other hand, optimization-based systems adjust optimization for fast adaptation [5, 18, 16].

Among optimization-based systems, MAML [5] has recently received interests, 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 quickly learn and achieve good generalization performance over tasks on average. While MAML boasts the simplicity, it shows relatively low performance on few-shot learning.

There has been several works that tried to improve the performance, especially on few-shot classification [1, 10, 12, 35, 8]. However, none of these methods tackles the problem

with the sharing of the starting point of adaptation to different tasks. Recently, there has been a few works [17, 21, 33] that try to achieve task-wise model or initialization through their proposed task embeddings. The metric-based system has a similar issue with MAML, and thus TADAM [17] proposes to learn task embeddings, which are then used to generate affine transformation parameters that transform the features.

In this work, we focus on analyzing the problems of MAML and improving its performance, while maintaining its generalizability. LEO [21] tries to solve the issue with the shared initialization by learning task embeddings through relation network, which are then used to generate input-dependent initializations in low-dimensional latent space. Another work that tries to relax the constraint on sharing the initialization is Multimodal MAML [33], where they propose to learn task embeddings and transform the MAML initialization with affine parameters.

In contrast to [33, 21] that only focus on making the initialization task-dependent, we approach the problem from the perspective of optimization and provide a new insight that 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 and is illustrated by 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. In fact, a large portion of the performance boost comes from the attenuation, not from the task-conditioned transformation (see Table 4).

From the perspective of optimization, we also provide more effective and efficient task embedding. Previous works [21, 33] try to achieve task-wise initializations through learning task embeddings directly from the input. However, learning such task embeddings without any task label is difficult and require specialized techniques, such as relation network [21] and metric learning [17] that may not be applicable in other complex problems such as in reinforcement learning. We argue and observe that the amount of *conflicts* varies among tasks, hinting that *conflicts* can be used to identify tasks. Since *conflicts* between the desired initialization by task and the learned initialization can be described with gradients (see Section 3.3), we demonstrate that gradients itself give task-specific optimization information and thus can be used to represent tasks. Because gradients are easily obtainable and model-agnostic, not only do we achieve effective task-wise initialization but also manage to maintain the simplicity and generalizability of MAML.

Overall, our proposed method greatly improves the performance of MAML while managing to maintain the simplicity and generalizability of MAML. Owing to its generalizability, we further show that not only does our method demonstrate a consistent improvement across domains, including reinforcement learning; but also our method can be easily applied to other MAML-based methods.

3. Proposed Method

3.1. Problem Formulation

Before introducing the proposed method, we start with the formulation of a generic meta-learning algorithm. We assume there is a distribution of tasks $p(\mathcal{T})$, from which meta-learning algorithm aims to learn the prior knowledge, represented by a model with parameters θ. Tasks, each of which is sampled from $p(\mathcal{T})$, are split into three disjoint sets: meta-training set, meta-validation set, and meta-test set. In k-shot learning, a task $\mathcal{T}_i$ is first sampled from the meta-training set, followed by sampling k number of examples $\mathcal{D}_{\mathcal{T}_i}$ from $\mathcal{T}_i$. These k examples are then used to quickly adapt a model with parameters, θ. Then, new examples $\mathcal{D}^{'}_{\mathcal{T}_i}$ are sampled from the same task $\mathcal{T}_i$ to evaluate the generalization performance on unseen examples with the corresponding loss function, $\mathcal{L}_{\mathcal{T}_i}$. The feedback from the loss is then used to adjust the model parameters θ to achieve better generalization. Finally, the meta-validation set is used for model selection, while the meta-test set is used for the final evaluation on the selected model.

3.2. Model-Agnostic Meta-Learning

To tackle the problem of fast adaptation to unseen tasks with few examples, we borrow the philosophy and the methodology from MAML [5]. MAML encodes prior knowledge in an initialization and seeks for a "good" common initial set of values for weights of a neural network across tasks. Formally, given a network f_θ with weights θ, MAML learns a set of initial weight values, θ, which will serve as a good starting point for fast adaptation to a new task $\mathcal{T}_i$, sampled from a task distribution $p(\mathcal{T})$. Given few examples $\mathcal{D}_{\mathcal{T}_i}$ and a loss function $\mathcal{L}_{\mathcal{T}_i}$ from the task $\mathcal{T}_i$, the network weights are adapted to $\mathcal{T}_i$ during inner-loop update as follows:

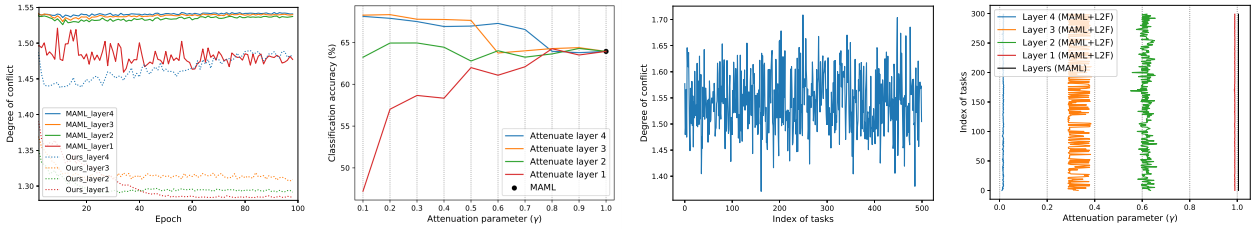
$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_\theta). \tag{1}$$

To give feedback on the generalization performance of the model with adapted weights θ_i' to each task, the model is evaluated on new examples, $\mathcal{D}^{'}_{\mathcal{T}_i}$ sampled from the same task $\mathcal{T}_i$. The feedback, manifested in the form of loss gradients, is used to update the initialization θ so that better generalization is achieved:

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

3.3. Definition of Conflict

While MAML is elegantly simple, its limitation comes from the very fact that the initialization is shared across a distribution of tasks. Despite the goal of MAML, which



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

Figure 3: Analysis on degree of conflict and attenuation: (a) Throughout training, degree of conflict is measured and observed to vary among layers. For MAML, deeper layers exhibit greater extent of conflict, which aligns with the observation that deeper layers encode more task specific features [34]. After applying L2F to MAML, conflict is observed to have decreased greatly. (b) Manual attenuation of an initialization by different levels (the lower γ, the stronger attenuation) for each layer affects the classification accuracy of a 4-layer CNN on miniImageNet. The figure suggests that deeper layers prefer the stronger attenuation. This supports our argument that the larger degree of conflict suggests the initialization quality is more compromised and that the compromised part needs to be minimized. (c) The degree of conflict between each meta-train task and the MAML initialization is observed to vary. This indicates that the amount of prior knowledge that is useful is different for each task. (d) Different attenuation parameters γ are generated by the proposed method for each meta-test task, especially for middle-level layers. This suggests that the degree of conflict varies for each task, especially in middle-level layers.

is to learn a "good" starting point for fast adaptation to new tasks, the shared initialization, in fact, hinders the fast learning process. This is illustrated by sharp optimization landscape during fast adaptation in Figure 2. This is mainly due to disagreement between tasks on the location of a "good" starting point. We call such disagreement *conflict*.

At each training iteration, each task $\mathcal{T}_i$ takes the initialization closer to the desired location via gradient: $\boldsymbol{u}_i = -\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}'}(f_{\theta_i'})$ during meta-update. However, since MAML shares the initialization, the update is made via gradients accumulated over a batch of tasks $\sum_i \boldsymbol{u_i}$ as in Equation (2). Hence, in the example of two tasks, the *conflict* occurs between tasks $\mathcal{T}_i$ and $\mathcal{T}_j$ when their gradient directions, i.e. directions of $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$, differ. The more their directions differ, the more the initialization update diverges from $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$, pointing towards the location that is not desirable for both $\mathcal{T}_i$ and $\mathcal{T}_j$. We refer to this phenomenon as *compromise in the initialization*.

We define *the degree of conflict* among tasks to be the average angle between $\boldsymbol{u}_i$ and $\sum_i \boldsymbol{u}_i$, which is measured as the average absolute arccosine of the dot product of the normalized vectors, $\mathbb{E}_{\mathcal{T}_i \sim p(T)}\left[\left|cos^{-1}(\hat{\boldsymbol{u}}_i \cdot \boldsymbol{v})\right|\right]$, where $\hat{\boldsymbol{u}}_i$ is $\frac{\boldsymbol{u}_i}{\|\boldsymbol{u}_i\|}$ and $\boldsymbol{v}$ is $\frac{\sum_i \boldsymbol{u}_i}{\left\|\sum_i \boldsymbol{u}_i\right\|}$. Figure 3(a) measures the *degree of conflict* at each epoch and demonstrates that the *conflict* is indeed more prominent in deeper layers, which aligns with the observation that the deeper layers encode more task-specific features [34].

3.4. Learning to Forget

When *the degree of conflict* is high, we say the initialization is more *compromised*, and hence the more difficult it is to learn new tasks quickly, as illustrated by sharp loss landscape in Figure 2. This suggests that the learner finds some part of the initialization to be irrelevant or even detrimental for learning a given task. We thus propose to discard such compromised part of the prior knowledge via attenuating the initialization parameters θ directly. Then, one may ask which parameter is compromised?

To answer the question, we refer to the previous finding that lower layers of a CNN encode general knowledge while deeper layers contain more task-specific information [34]. Upon this observation, we hypothesize that lower layers do not need much attenuation while deeper layers do. To support our hypothesis, we perform an experiment, shown in Figure 3(b), where we vary the amount of attenuation (γ^j) on each layer to observe how much each layer benefits. As expected, deeper layers favor stronger attenuation while lower layers prefer little to no attenuation. This leads to the second question: How much should the parameters be attenuated layer-wise?

One answer would be to let a model learn to find an optimal set of attenuations. The answers to these two questions lead to our proposal: learn layer-wise attenuation via applying a single learnable parameter γ^j on the initialization parameters of each layer θ^j as follows:

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

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. Although this may reduce the extent of compromise that may exist in the original MAML initialization, one may ask if the amount of unnecessary or contradicting information in the initialization is equal across

Algorithm 1 Proposed Meta-Learning

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 $(\mathcal{D}_{\mathcal{T}_i}, \mathcal{D}'_{\mathcal{T}_i})$ from $\mathcal{T}_i$
    Compute $\mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_\theta)$ by evaluating $\mathcal{L}_{\mathcal{T}_i}$ with respect to $\mathcal{D}_{\mathcal{T}_i}$
    Compute attenuation parameter γ for each layer: $\{\gamma_i^j\}^{j=1...l} = g_\phi(\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_\theta))$,
    Compute attenuated 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 $\mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_{\theta'_i})$ by evaluating $\mathcal{L}_{\mathcal{T}_i}$ with respect to $\mathcal{D}_{\mathcal{T}_i}$
      Perform gradient descent to compute adapted weights: $\theta'_i = \theta'_i - \alpha \nabla_{\theta'_i} \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_{\theta'_i})$
    end for
    Compute $\mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}'_{\mathcal{T}_i}}(f_{\theta'_i})$ by evaluating $\mathcal{L}_{\mathcal{T}_i}$ with respect to $\mathcal{D}'_{\mathcal{T}_i}$
  end for
  Perform gradient descent to update weights: $(\theta, \phi) \leftarrow (\theta, \phi) - \eta \nabla_{(\theta,\phi)} \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}'_{\mathcal{T}_i}}(f_{\theta'_i})$
end while

tasks.

Surely, the degree of agreement and disagreement with others differs for different tasks. This can be observed in Figure 3(c), where the measured degree of conflict is observed to vary for each task. As a result, 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 such conflict, in addition to the layer-wise attenuation, we propose a task-dependent attenuation. But, this poses another question: What information can be used to make attenuation task-dependent?

We turn to gradients $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_\theta)$ for the answer. Gradients, used for fast adaptation via 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 $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_\theta)$ at the initialization and condition a network g_ϕ on it to generate the task-dependent attenuation:

$$\gamma_i = g_\phi(\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\mathcal{D}_{\mathcal{T}_i}}(f_\theta)), \tag{4}$$

where $\gamma_i = \{\gamma_i^j\}$ is the set of layer-wise gammas for the i-th task and g_ϕ is a 3-layer MLP network of parameters ϕ, with a sigmoid at the end to facilitate attenuation. For the network g_ϕ to generate layer-wise gammas, the network is conditioned on the layer-wise mean of gradients.

After the initialization is adapted to each task, the network undergoes fast adaptation as in Equation (1) and the initialization is updated as in Equation (2) during training. 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 on various problems, including few-shot classification, regression, and reinforcement learning.

4.1. Few-Shot Classification

Two well-known datasets, miniImageNet and tieredImageNet are used for the classification test, both of which are extracted from ImageNet dataset while taking into account for few-learning scenarios. miniImageNet is constructed by randomly selecting 100 classes from the ILSVRC-12 dataset, with each class consisting of 600 images of size 84 $\times$ 84 [32]. The constructed dataset is divided into 3 disjoint subsets: 64 classes for training, 16 for validation, and 20 for test as in [18].

tieredImageNet is a larger subset with 608 classes with 779,165 images of size 84 $\times$ 84 in total. Classes are grouped into 34 categories, according to ImageNet hierarchy. These categories are then split into 3 disjoint sets: 20 categories for training, 6 for validation, and 8 for test. According to [20], this minimizes class similarity between training and test and thus makes the problem more challenging and realistic. Experiments for tieredImageNet and miniImageNet are conducted under typical settings: 5-way 1-shot and 5-way

	Backbone	miniImageNet	
		1-shot	5-shot
Matching Network [32]	4 conv	$43.44 \pm 0.77\%$	$55.31 \pm 0.73\%$
Meta-Learner LSTM (Ravi et al. 2017)	4 conv	$43.56 \pm 0.84\%$	$60.60 \pm 0.71\%$
MetaNet (Munkhdalai et al. 2017)	5 conv	$49.21 \pm 0.96\%$	–
LLAMA [6]	4 conv	$49.40 \pm 0.84\%$	–
Relation Network [28]	4 conv	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$
Prototypical Network (Snell et al. 2017)	4 conv	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
MAML (Finn et al. 2017)	4 conv	$48.70 \pm 1.75\%$	$63.11 \pm 0.91\%$
MAML++ (Antoniou et al. 2019)	4 conv	$52.15 \pm 0.26\%$	$68.32 \pm 0.44\%$
MAML+L2F (Ours)	4 conv	$52.10 \pm 0.50\%$	$69.38 \pm 0.46\%$
MetaGAN [35]	ResNet12	$52.71 \pm 0.64\%$	$68.63 \pm 0.67\%$
SNAIL [13]	ResNet12*	$55.71 \pm 0.99\%$	$68.88 \pm 0.92\%$
adaResNet [15]	ResNet12	$56.88 \pm 0.62\%$	$71.94 \pm 0.57\%$
CAML [8]	ResNet12*	$59.23 \pm 0.99\%$	$72.35 \pm 0.71\%$
TADAM (Oreshkin et al. 2018)	ResNet12*	$58.5 \pm 0.3\%$	$76.7 \pm 0.3\%$
MAML	ResNet12	$51.03 \pm 0.50\%$	$68.26 \pm 0.47\%$
MAML+L2F (Ours)	ResNet12	$57.48 \pm 0.49\%$	$74.68 \pm 0.43\%$
LEO [21]	WRN34*	$61.76 \pm 0.08\%$	$77.59 \pm 0.12\%$
LEO (reproduced)	WRN34*	$61.50 \pm 0.17\%$	$77.12 \pm 0.07\%$
LEO+L2F (Ours)	WRN34*	$\mathbf{62.12 \pm 0.13\%}$	$\mathbf{78.13 \pm 0.15\%}$

* a pre-trained network.

Table 1: Test accuracy on 5-way miniImageNet classification

	Backbone	tieredImageNet	
		1-shot	5-shot
MAML	4 conv	$49.06 \pm 0.50\%$	$67.48 \pm 0.47\%$
MAML+L2F (Ours)	4 conv	$54.40 \pm 0.50\%$	$73.34 \pm 0.44\%$
MAML	ResNet12	$58.58 \pm 0.49\%$	$71.24 \pm 0.43\%$
MAML+L2F (Ours)	ResNet12	$63.94 \pm 0.48\%$	$77.61 \pm 0.41\%$
LEO	WRN34*	$66.33 \pm 0.05\%$	$81.44 \pm 0.09\%$
LEO (reproduced)	WRN34*	$67.02 \pm 0.11\%$	$82.29 \pm 0.16\%$
LEO+L2F (Ours)	WRN34*	$\mathbf{68.00 \pm 0.11\%}$	$\mathbf{83.02 \pm 0.08\%}$

* a pre-trained network.

Table 2: Test accuracy on 5-way tieredImageNet classification

5-shot classification. For more experiments on other datasets, such as FC100 [17], CIFAR-FS [3], and Meta-Dataset [31], please see the supplementary materials.

4.1.1 Results

The results of our proposed approach, other baselines and existing state-of-the-art approaches on the miniImageNet and tieredImageNet are presented in Table 1 and Table 2, respectively. The proposed method improves MAML by a large margin. We note that our proposed approach remains model-agnostic and achieves better or comparable accuracy to the state-of-the-art approaches with the same backbone, even without fine-tuning. To show generalization of the contribution, we apply L2F to the state-of-the-art MAML-based system LEO and demonstrate the performance improvement, achieving the new state-of-the-art performance.

4.1.2 Ablation Studies

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	$64.54 \pm 0.46\%$	–

Table 3: Ablation studies on inner-loop update steps on 5-way 5-shot miniImageNet classification.

Inner-loop update steps One may argue that the comparisons are not fair because there is one extra adjustment to initialization parameters before inner-loop updates. Table 3 shows ablation studies on the number of inner-loop updates for the proposed and the baseline to demonstrate that the performance gain is not due to an extra number of adjustments to parameters. Rather, the benefits come from *forgetting* the unnecessary information, helping the learner quickly adapt to new tasks.

Attenuation Scope One may be curious and ask: Is layer-wise attenuation the best way to go? Thus, we analyze different scopes of attenuation; a single attenuation parameter for the whole network, or an individual attenuation parameter for each layer, each filter, and each weight of the network. To focus on investigating which scope of attenuation is most beneficial, we remove the task-dependent part and make the attenuation parameters learnable (with values initialized to be 1), rather than generated by the network g_ϕ.

We perform an ablation study with a 4-layer CNN in 5-way 5-shot classification setting on miniImageNet and present results in Table 4. As expected, the layer-wise attenuation gave the most performance gain. Weight-wise or filter-wise attenuation parameter may have finer control, but these parameters have limited scope in that they do not have information about conflicts that occur at the level of layers or network. On the other hand, layer-wise and network-wise parameters gain information about conflicts in neighbor weights as gradients pass through different weights/filters to reach the same attenuation parameter, since the attenuation parameter is shared by these weights/filters. In the meantime, network-wise parameters do not have enough control and thus perform worse than the layer-wise parameters. In the trade-off between control and information gain, layer-wise has shown to strike the right balance.

Effect of Task-Conditioning Table 4 reports lower performance of layer-wise attenuation model, compared to our full model, MAML+L2F. The only difference between the layer-wise attenuation model and ours is that the layer-wise attenuation model lacks the task-conditioning. One can observe that the most performance gain in our method comes from the attenuation, alluding to the importance of attenuation. Regardless, the task-conditioning does improve the performance as well.

Representation of Task Embedding To verify that gradients contain high-quality information about tasks, we condition the network g on the mean of class prototypes from the pre-trained prototypical network [27](similar to TADAM [17]) as task representation. Table 5 demonstrates that our method with gradients as task representation performs similarly or slightly better than the one with the mean of class prototypes. This exhibits the effectiveness of gra-

Attenuation Scope	Accuracy
None (MAML, our reproduction)	$63.94 \pm 0.48\%$
parameter-wise	$64.7 \pm 0.43\%$
filter-wise	$65.35 \pm 0.48\%$
layer-wise	$\mathbf{68.49 \pm 0.41\%}$
network-wise	$67.84 \pm 0.46\%$
MAML+L2F (Ours)	$\mathbf{69.38 \pm 0.46\%}$

Table 4: Ablation studies on attenuation scope. Except MAML+L2F, all models learn task-independent attenuation parameters to illustrate the effect of attenuation scope alone, without task-conditioning.

	miniImageNet 5-shot
Features (class prototype)	$68.73 \pm 0.46\%$
Gradients (Ours, MAML+L2F)	$69.48 \pm 0.46\%$

Table 5: Ablation studies on types of representation for task embedding

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

Table 6: Ablation studies on task-conditioned transformation to illustrate the effectiveness of attenuation.

dients as task representation from the perspective of the optimization, especially because gradients are simple to obtain and model-agnostic while class prototypes are high-dimensional and not applicable across different domains.

Effect of Attenuation To analyze how much performance gain comes from each part of L2F (i.e. *forgetting* and task-dependency), we apply each module separately to MAML and present results in Table 6. Since the investigation on effectiveness of task-dependency has already been presented in Table 4, we now focus on the effectiveness of the attenuation, compared to other variant transformations. To that end, we explore different types of task-dependent transformations of the initialization. We start with the simple superset of the attenuation: γ 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: affine transformation (Model 2), where 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 6 illustrates that MAML gains performance boost throughout different types of task-dependent transformation, suggesting the benefits of the task-dependency. It is reasonable to expect that more flexibility of transformation (Model 2 and 3) would allow for tasks to bring the initialization to more appropriate location for fast adaptation. Interestingly, the classification accuracy drops as more flexibility is given to the transformation of the initialization. This seeming contradiction underlines the necessity of attenuation (sigmoided γ_i^j in our model), rather than just naïve transformation, of the initialization to *forget* the compromised part of the prior knowledge encoded in the initialization.

We would like to stress that MAML with task-independent layer- or network-wise attenuation in Table 4 performs better than other task-conditioned transformations in Table 6. This suggests that it is more important to *forget* the compromised initialization than making it task-adaptive.

	Models	1 step	2 steps	5 steps
5-shot training	MAML	1.2247	1.0268	0.8995
	MAML+L2F (Ours)	**1.0537**	**0.8426**	**0.7096**
10-shot training	MAML	0.9884	0.6192	0.4072
	MAML+L2F (Ours)	**0.8069**	**0.5317**	**0.3696**
20-shot training	MAML	0.6144	0.3346	0.1817
	MAML+L2F (Ours)	**0.5475**	**0.2805**	**0.1629**

Table 7: MSE averaged over the sampled 100 points with 95%confidence intervals on k-shot regression. Our method consistently outperforms across all gradient steps.

4.2. Regression

We investigate the generalizability of the proposed method across domains, starting with evaluating the performance in k-shot regression. In k-shot regression, the objective is to fit a function, given k samples of points. Following the general settings from [5, 12], the target function is set to be a sinusoid with varying amplitude and phase between tasks. The sampling range of amplitude, frequency, and phase defines a task distribution and is set to be the same for both training and evaluation. Regression is visualized in Figure 4(a), while its prediction, measured in mean-square error (MSE), is presented in Table 7. The results demonstrate that our method not only converges faster but also fits to target functions more accurately.

To further stress the generalization of the MAML+L2F initialization, we extensively increase the degree of conflicts between new tasks and the prior knowledge. To that end, we modify the setting such that amplitude, frequency, and phase are sampled from the non-overlapped ranges for training and evaluation (please refer to the supplementary material for details). In Figure 4(b), our model exhibits higher accuracy and thus claims the better generalization.

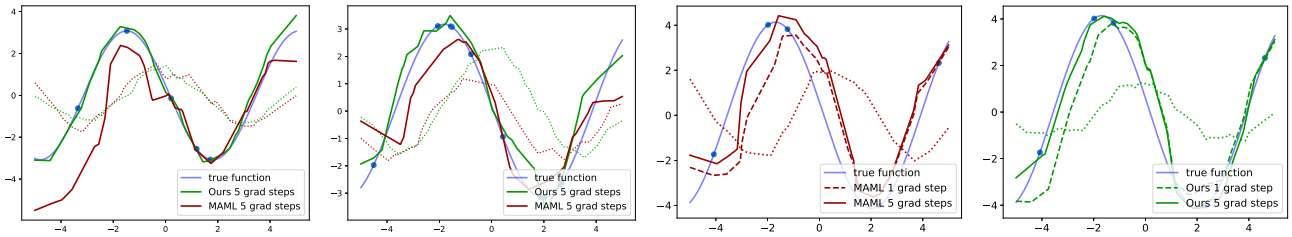
4.3. Reinforcement Learning

To further validate the generalizability of L2F, we evaluate the performance in reinforcement learning, specifically in 2D navigation and locomotion environments from [4] as in [5]. We briefly outline the task description below (please refer to the supplementary material for details). Figure 5 presents consistent improvement over MAML across different experiments. This solidifies the generalizability and effectiveness of our proposed method.

4.3.1 2D Navigation

A 2D navigation task is to move an agent from the starting point to the destination point in 2D space, where the reward is defined as the negative of the squared distance to the destination point. We follow the experiment procedure from [5], where they fix the starting point and only vary the location of destination between tasks.

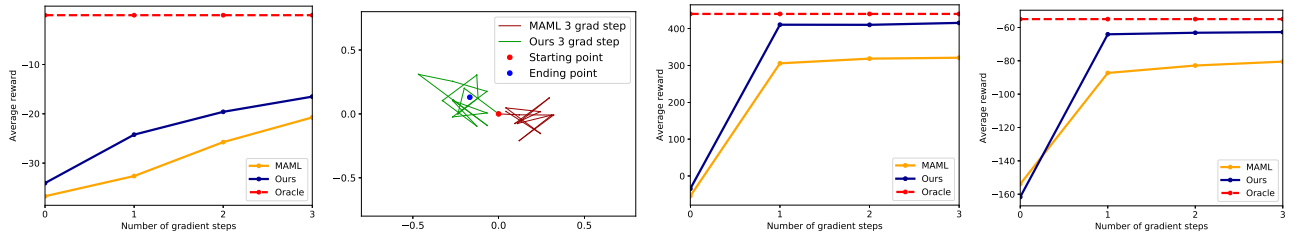
Figure 5(a) presents faster and more precise navigation by our model in both experiment settings, both quantitatively



(a) 5, 10-shot regression

(b) Non-overlapped task distributions on 5-shot

Figure 4: MAML + L2F (Ours) vs MAML on Few-shot regression: (a) Tasks are sampled from the same distribution for training and evaluation. (b) Tasks are sampled from the non-overlapped distributions for training and evaluation. In both cases, MAML+L2F (Ours) is more fitted to the true function.



(a) 2D navigation

(b) Half-cheetah (direction)

(c) Half-cheetah (velocity)

Figure 5: Reinforcement learning results for 3 different environments. The results show that MAML+L2F(Ours) can adapt to each task substantially faster than MAML.

and qualitatively. This solidifies the severity of the conflicts between tasks.

4.3.2 Mujoco

As a more complex reinforcement-learning environment, we experiment on locomotion with the MuJoCo simulator [30], where there are two sets of tasks: a robot is required to move in a particular direction in one set and move with a particular velocity in the other. For both experiments,our method outperforms MAML in large margins as shown in Figure 5(b), (c).

4.4. Loss Landscape

We further validate the effectiveness of our model by illustrating the smoother loss landscape after applying L2F to MAML for the miniImageNet classification tasks, as shown in Figure 2. At the initial stages of training, L2F appears to struggle more, while optimization of MAML seems more stable. This may seem contradictory at first but this actually validates our argument about conflicts between tasks even further. At the beginning, the MAML initialization is not trained enough and thus does not have sufficient prior knowledge of task distribution yet. As training proceeds, the initialization encodes more information about task distribution and encounters conflicts between tasks more frequently. As for L2F, the attenuator g_ϕ initially does not have enough knowledge about the task distribution and thus generates meaningless attenuation γ_i, deteriorating the initialization. But, the attenuator increasingly encodes more information about the task distribution, generating more appropriate attenuation γ_i that corresponds to tasks well. The generated γ_i accordingly allows for a learner to *forget* the irrelevant part of prior knowledge to help fast adaptation, as illustrated by increasing stability and smoothness of landscape.

5. Conclusion

In this paper, we argue that forcibly sharing a common initialization in MAML induces conflicts across tasks and thus results in the compromised location of the initialization. The severely sharp loss landscape asserts that such compromise makes the MAML initialization a "bad" starting position for fast adaptation. We propose to resolve this discrepancy by facilitating *forgetting* (attenuating) the irrelevant information that may hinder fast adaptation. Specifically, we propose a task-dependent layer-wise attenuation, named L2F, motivated by the observation that the degree of compromise varies between network layers and tasks. Through extensive experiments across different domains, we validate our argument that selective *forgetting* greatly facilitates fast adaptation while retaining the simplicity and generalizability of MAML.

Acknowledgements This work was supported by IITP grant funded by the Ministry of Science and ICT of Korea (No. 2017-0-01780), and Hyundai Motor Group through HMG-SNU AI Consortium fund (No. 5264-20190101).

References

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. In *ICLR*, 2019. 2

[2] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pages 6–8. Univ. of Texas, 1992. 2

[3] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019. 6

[4] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *ICML*, 2016. 7

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1, 2, 3, 7

[6] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *ICLR*, 2018. 5

[7] Sepp Hochreiter, A Younger, and Peter Conwell. Learning to learn using gradient descent. *Artificial Neural Networks, ICANN 2001*, pages 87–94, 2001. 2

[8] Xiang Jiang, Mohammad Havaei, Farshid Varno, Gabriel Chartrand, Nicolas Chapados, and Stan Matwin. Learning to learn with conditional class dependencies. In *ICLR*, 2019. 2, 5

[9] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, 2015. 2

[10] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *ICML*, 2018. 2

[11] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *NIPS*, 2018. 2

[12] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017. 2, 7

[13] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018. 5

[14] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, 2017. 2

[15] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *ICML*, 2018. 2, 5

[16] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 2

[17] Boris N. Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NIPS*, 2018. 2, 3, 6

[18] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2, 5

[19] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshop*, 2014. 1

[20] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018. 5

[21] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 3, 5

[22] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICLR*, 2016. 2

[23] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *NIPS*, 2018. 2

[24] Jurgen 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. 2

[25] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992. 2

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[27] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 2, 6

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2, 5

[29] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 2

[30] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012. 8

[31] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *CoRR*, abs/1903.03096, 2019. 1, 6

[32] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, 2016. 2, 5

[33] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Toward multimodal model-agnostic meta-learning. In *NIPS Meta-Learning Workshop*, 2018. 3

[34] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2, 4

[35] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *NIPS*, 2018. 2, 5