Not All Tasks are Equal - Task Attended Meta-learning for Few-shot Learning

Anonymous authors
Paper under double-blind review

Abstract

Meta-learning (ML) has emerged as a promising direction in learning models under constrained resource settings like few-shot learning. The popular approaches for ML either learn a generalizable initial model or a generic parametric optimizer through episodic training. The former approaches leverage the knowledge from a batch of tasks to learn an optimal prior. In this work, we study the importance of tasks in a batch for ML. We hypothesize that the common assumption in batch episodic training where each task in a batch has an equal contribution to learning an optimal meta-model need not be true. We propose to weight the tasks in a batch according to their "importance" in improving the meta-model's learning. To this end, we introduce a training curriculum called task attended meta-training to learn a meta-model from weighted tasks in a batch. The task attention module is a standalone unit and can be integrated with any batch episodic training regimen. The comparisons of the task-attended ML models with their non-task-attended counterparts on complex datasets like miniImagenet, FC100 and tieredImagenet validate its effectiveness.

1 Introduction

The ability to infer knowledge and discover complex representations from data has made deep learning models widely popular in the machine learning community. However, these models are data-hungry, often requiring large volumes of labeled data for training. Collection and annotation of such large amounts of training data may not be feasible for many real life applications, especially in domains that are inherently data constrained, like medical and satellite image classification, drug toxicity estimation, etc. Meta-learning (ML) has emerged as a promising direction for learning models in such settings, where only a limited amount (few-shots) of labeled training data is available. A typical ML algorithm employs an episodic training regimen that differs from the training procedure of conventional learning tasks. This episodic meta-training regimen is backed by the assumption that a machine learning model quickly generalizes to novel unseen data with minimal fine-tuning when trained and tested under similar circumstances Vinyals et al. (2016). To facilitate such a generalization capacity, a meta-training phase is undertaken, where the model is trained to optimize its performance on several homogeneous tasks/episodes randomly sampled from a dataset. Each episode or task is a learning problem in itself. In the few-shot setting each task is a classification problem, a collection of K support (train) and Q query (test) samples corresponding to each of the N classes. Task-specific knowledge is learned using the support data, and meta-knowledge across the tasks is learned using query samples, which essentially encodes "how to learn a new task effectively."

The learned meta-knowledge is generic and agnostic to tasks from the same distribution. It is typically characterized in two different forms - either as an optimal initialization for the machine learning model or a learned parametric optimizer. Under the optimal initialization view, the learned meta-knowledge represents an optimal prior over the model parameters, that is equidistant, but close to the optimal parameters for all individual tasks. This enables the model to rapidly adapt to unseen tasks from the same distribution Finn et al. (2017); Li et al. (2017); Jamal & Qi (2019). Under the parametric optimizer view, meta-knowledge pertaining to the traversal of the loss surface of individual tasks is learned by the meta-optimizer. Through learning task specific and task agnostic characteristics of the loss surface, a parametric optimizer can thus effectively guide the base model to traverse the loss surface and achieve superior performance on unseen tasks from the same distribution Ravi & Larochelle (2017).

Initialization based ML approaches accumulate the meta-knowledge by simultaneously optimizing over a batch of tasks. On the other hand, a parametric optimizer sequentially accumulates meta-knowledge across individual tasks. The sequential accumulation process leads to a long oscillatory optimization trajectory and a bias towards the last task, limiting the parametric optimizer's task agnostic potential. Leveraging common knowledge from a batch of tasks to learn the parametric optimizer can help address this issue. We first propose to accumulate meta-knowledge in a batch mode for the parametric optimizer. Further, under such batch episodic training (for both initialization and optimization views), a common assumption in ML that the randomly sampled episodes of a batch contribute equally to improving the learned meta-knowledge need not hold good. Due to the latent properties of the sampled tasks in a batch and the model configuration, some tasks may be better aligned with the optimal meta-knowledge than others. We hypothesize that proportioning the contribution of a task as per its alignment towards the optimal meta-knowledge can improve the meta-model's learning. This is analogous to classical machine learning algorithms like sample re-weighting, which however, operate at sample granularity. In re-weighting, samples leading to false positives are prioritized and therefore replayed. Hence, the latent properties due to which a sample is prioritized are explicitly defined. For complex task distributions, explicitly handcrafting the notion of "importance" of a task would be hard.

To this end, we propose a task attended meta-training curriculum that employs an attention module that learns to assign weights to the tasks of a batch with experience. The attention module is parametrized as a neural network that takes meta-information in terms of the model's performance on the tasks in a batch as input and learns to associate weights to each of the tasks according to their contribution in improving the meta-model. Overall, we make the following contributions,

- We propose a task attended meta-training strategy wherein different tasks of a batch are weighted according to their "importance" defined by the attention module. This attention module is a standalone unit that can be integrated into any batch episodic training regimen.
- To integrate task attention module with the parametric optimizer, we design a batch-mode parametric optimizer (MetaLSTM++) and experimentally show its merit on miniImagenet, FC100, and tieredImagenet datasets.
- We conduct extensive experiments on miniImagenet, FC100, and tieredImagenet datasets and compare ML algorithms like MAML, MetaSGD, ANIL, and MetaLSTM++ with their non-task-attended counterparts to validate the effectiveness of the task attention module and its coupling with any batch episodic training regimen.
- We also perform exhaustive empirical analysis and visual inspections to decipher the working of the task attention module.

2 Related Work

ML literature is profoundly diverse and may broadly be classified into *initialization* Finn et al. (2017); Li et al. (2017); Jamal & Qi (2019); Raghu et al. (2020); Rusu et al. (2019); Sun et al. (2019) and *optimization approaches* Ravi & Larochelle (2017); Aimen et al. (2021) depending on the metaknowledge. However, these approaches assume uniform contribution of tasks in learning a meta-model. In supervised learning, assigning non-uniform priorities to the samples is not new Kahn & Marshall (1953); Shrivastava et al. (2016). Self-paced learning Kumar et al. (2010) and hard example mining Shrivastava et al. (2016) have popularly been used to reweight the samples and various attributes like losses, gradients, and uncertainty have been used to assign priorities to samples Lin et al. (2017); Zhao & Zhang (2015); Chang et al. (2017). Zhao & Zhang (2015) introduce importance sampling to reduce variance and improve the convergence rate of stochastic optimization algorithms over uniform sampling. They theoretically prove that the reduction in the variance is possible if the sampling distribution depends on the norm of the gradients of the loss function. Chang et al. (2017) conclude that mini-batch SGD for classification is improved by emphasizing the uncertain examples.

Lin et al. (2017) propose reshaped cross-entropy loss (focal loss) that down-weights the loss of confidently classified samples. Nevertheless, assigning non-uniform priorities to tasks in meta-learning is under-explored and has recently drawn attention Kaddour et al. (2020); Gutierrez & Leonetti (2020); Liu et al. (2020); Yao et al. (2021). Gutierrez & Leonetti (2020) propose Information-Theoretic Task Selection (ITTS) algorithm to filter training tasks that are distinct from each other and close to the tasks of the target distribution. However, this algorithm results in a smaller pool of training tasks. A model trained on the smaller subset learns better than the one trained on the original set. On the other hand, Kaddour et al. (2020) propose probabilistic active meta-learning (PAML) that learns probabilistic task embeddings. Scores are assigned to these embeddings to select the next task presented to the model. These algorithms are, however, specific to meta-reinforcement learning (meta-RL). On the contrary, our focus is on the few shot classification problem. Liu et al. (2020) propose a greedy class-pair potential-based adaptive task sampling strategy wherein task selection depends on the difficulty of all class-pairs in a task. This sampling technique is static and operates at a class granularity. On the other hand, our approach is dynamic and operates at a task granularity. Assigning non-uniform weights to samples prevents overfitting on corrupt data points Ren et al. (2018b); Jiang et al. (2018). Ren et al. (2018b) used gradient directions to re-weight the data points, and Jiang et al. (2018) learned a curriculum on examples using a mentor network. However, these approaches assume availability of abundant labeled data. Yao et al. (2021) extended Jiang et al. (2018) to few-shot learning setup. They propose a neural schedular to predict the sampling probability of tasks in a candidate pool. Parallel to Jiang et al. (2018), they consider noisy and imbalanced task distributions. Our work is different from these approaches as we do not propose a task sampling strategy but a dynamic task-batch re-weighting mechanism for the meta-model update in a few-shot learning setup. Further, our problem is different and more generic as we consider prevalent task distributions which are neither noisy nor imbalanced. Contrary to our idea is TAML Jamal & Qi (2019) - a meta-training curriculum that enforces equity across the tasks in a batch. We show that weighting the tasks according to their "importance" and hence utilizing the diversity present in a batch given the meta-model's current configuration offers better performance than enforcing equity in a batch of tasks.

3 Preliminary

In a typical ML setting, the principal dataset $\mathcal{D}$ is divided into disjoint meta-sets $\mathcal{M}$ (meta-train set), $\mathcal{M}_v$ (meta-validation set) and $\mathcal{M}_t$ (meta-test set) for training the model, tuning its hyperparameters and evaluating its performance, respectively. Every meta-set is a collection of tasks $\mathcal{T}$ drawn from the joint task distribution $P(\mathcal{T})$ where each task $\mathcal{T}_i$ consists of support $D_i = \{\{x_k^c, y_k^c\}_{k=1}^{K}\}_{c=1}^{N}$ and query set $D_i^* = \{\{x_q^{*c}, y_q^{*c}\}_{q=1}^{Q}\}_{c=1}^{N}$. Here (x, y) represents a (sample, label) pair and N is the number of classes, K and Q are the number of samples belonging to each class in the support and query set, respectively. According to support-query characterization $\mathcal{M}$, $\mathcal{M}_v$ and $\mathcal{M}_t$ could be represented as $\{(D_i, D_i^*)\}_{i=1}^{M}$, $\{(D_i, D_i^*)\}_{i=1}^{R}$, $\{(D_i, D_i^*)\}_{i=1}^{S}$ where M, R and S are the total number of tasks in $\mathcal{M}$, $\mathcal{M}_v$ and $\mathcal{M}_t$ respectively. During meta-training on $\mathcal{M}$, meta-model θ is adapted on D_i of each $\mathcal{T}_i$ to ϕ_i. The adapted model ϕ_i is then evaluated on D_i^* to update θ. The output of this episodic training is either an optimal prior or a parametric optimizer, both aiming to facilitate the rapid adaptation of the model on unseen tasks from $\mathcal{M}_t$.

3.1 Meta-knowledge as an Optimal Initialization

When meta-knowledge is a generic initialization on the model parameters learned through the experience over various tasks, it is enforced to be close to each individual training tasks' optimal parameters. A model initialized with such an optimal prior quickly adapts to unseen tasks from the same distribution during meta-testing. **MAML** Finn et al. (2017) employs a nested iterative process to learn the task-agnostic optimal prior θ. In the inner iterations representing the task adaptation steps, θ is separately fine-tuned for each meta-training task $\mathcal{T}_i$ of a batch using D_i to obtain ϕ_i through gradient descent on the train loss L_i. Specifically, ϕ_i is initialized as θ and updated using $\phi_i \leftarrow \phi_i - \alpha \nabla_{\phi_i} L_i(\phi_i)$, T times resulting in the adapted model ϕ_i^T. In the outer loop, meta-knowledge is gathered by optimizing θ over loss L_i^* computed with the task adapted model parameters ϕ_i^T on query dataset D_i^*. Specifically, during meta-optimization $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^{B} L_i^*(\phi_i^T)$ using a task batch of size B. **MetaSGD** Li et al. (2017) improves upon MAML by learning parameter-specific learning rates α in addition to the optimal initialization in a similar nested iterative procedure. Meta-knowledge is gathered by optimizing θ and α in the outer loop using the loss L_i^* computed on query set D_i^*. Specifically, during meta-optimization $(\theta, \alpha) \leftarrow (\theta, \alpha) - \beta \nabla_{(\theta,\alpha)} \sum_{i=1}^{B} L_i^*(\phi_i^T)$. Learning dynamic learning rates for each parameter of a model makes MetaSGD faster and more generalizable than MAML. A single adaptation step is sufficient to adjust the model towards a new task. The performance of MAML is attributed to the reuse of the features across tasks rather than the rapid learning of new tasks Raghu et al. (2020). Exploiting this characteristic, **ANIL** freezes the feature backbone learned in the outer loop and only adapts the classifier in the inner loop T times. Specifically during adaptation $\phi_i^l \leftarrow \phi_i^l - \alpha \nabla_{\phi_i^l} L_i(\phi_i^l)$, where $1, \ldots, l$ are the layers of the model. During meta-optimization $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^{B} L_i^*([\phi_i^l]^T)$. Freezing the feature backbone during adaptation reduces the overhead of computing gradient through the gradient (differentiating through the inner loop), and thereby heavier backbones could be used for the feature extraction. **TAML** Jamal & Qi (2019) suggests that the optimal prior learned by MAML may still be biased towards some tasks. They propose to reduce this bias and enforce equity among the tasks by explicitly minimizing the inequality among the performances of tasks in a batch. The inequality defined using statistical measures such as Theil Index, Atkinson Index, Generalized Entropy Index, and Gini Coefficient among the performances of tasks in a batch is used as a regularizer while gathering the meta-knowledge. For the baseline comparison, in our experiments, we use the Theil index for TAML owing to its average best results. Specifically during meta-optimization $\theta \leftarrow \theta - \beta \nabla_\theta \left[\sum_{i=1}^{B} L_i^*(\phi_i^T) + \lambda \left\{ \frac{L_i^*(\theta)}{\bar{L}^*(\theta)} \ln \frac{L_i^*(\theta)}{\bar{L}^*(\theta)} \right\} \right]$ (for TAML-Theil Index) where B is the number of tasks in a batch, L_i^* is loss of task $\mathcal{T}_i$ on the query set D_i^* and $\bar{L}^*$ is the average test loss on a batch of tasks. As TAML enforces equity of the optimal prior towards meta-train tasks, it counters the adaptation, which leads to slow and unstable training largely dependent on λ.

3.2 Meta-knowledge as a Parametric Optimizer

A regulated gradient-based optimizer gathers the task-specific and task-agnostic meta-knowledge to traverse the loss surfaces of tasks in the meta-train set during meta-training. A base model guided by such a learned parametric optimizer quickly finds the way to minima even for unseen tasks sampled from the same distribution during meta-testing. **MetaLSTM** Ravi & Larochelle (2017) is a recurrent parametric optimizer θ that mimics the gradient-based optimization of a base model ϕ. This recurrent optimizer is an LSTM Hochreiter & Schmidhuber (1997) and is inherently capable of performing two-level learning due to its architecture. During adaptation of ϕ_i on D_i, θ takes meta information of ϕ_i characterized by its current loss L_i and gradients $\nabla_{\phi_i}(L_i)$ as input and outputs the next set of parameters for ϕ_i. This adaptation procedure is repeated T times resulting in the adapted base-model ϕ_i^T. Internally, the cell state of θ corresponds to ϕ_i, and the cell state update for θ resembles a learned and controlled gradient update. The emphasis on previous parameters and the current update is regulated by the learned forget and input gates respectively. While adapting ϕ_i to D_i, information about the trajectory on the loss surface across the adaptation steps is captured in the hidden states of θ, representing the task-specific knowledge. During meta-optimization, θ is updated based on the loss L_i^* of model computed on query set D_i^* to garner the meta-knowledge across tasks. Specifically, during meta-optimization, $\theta \leftarrow \theta - \beta \nabla_\theta L_i^*(\phi_i^T)$. A caveat of MetaLSTM is the sequential update to the parametric optimizer θ after adapting to each task. As a result, the parametric optimizer traverses the loss surface in an ordered sequence of task-specific optima. This leads to a longer and oscillatory optimization trajectory and bias of θ towards the final task. We propose to overcome this bottleneck by learning θ according to the training procedure of optimal initialization ML approaches, which we term as **MetaLSTM++** Aimen et al. (2021). Unlike MetaLSTM, the meta-knowledge θ of MetaLSTM++ is updated based on the average test loss of the tasks in a batch. This is intuitive as a batch of tasks may better approximate the task distribution than a single task. The batch update on θ makes the optimization trajectory smooth, short, and robust to task order. Specifically, during meta-optimization $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^{B} L_i^*(\phi_i^T)$.

4 Task Attention in Meta-learning

A common assumption under the batch-wise episodic training regimen adopted by ML is that each task in a batch has an equal contribution in improving the learned meta-knowledge. However, this need not always be

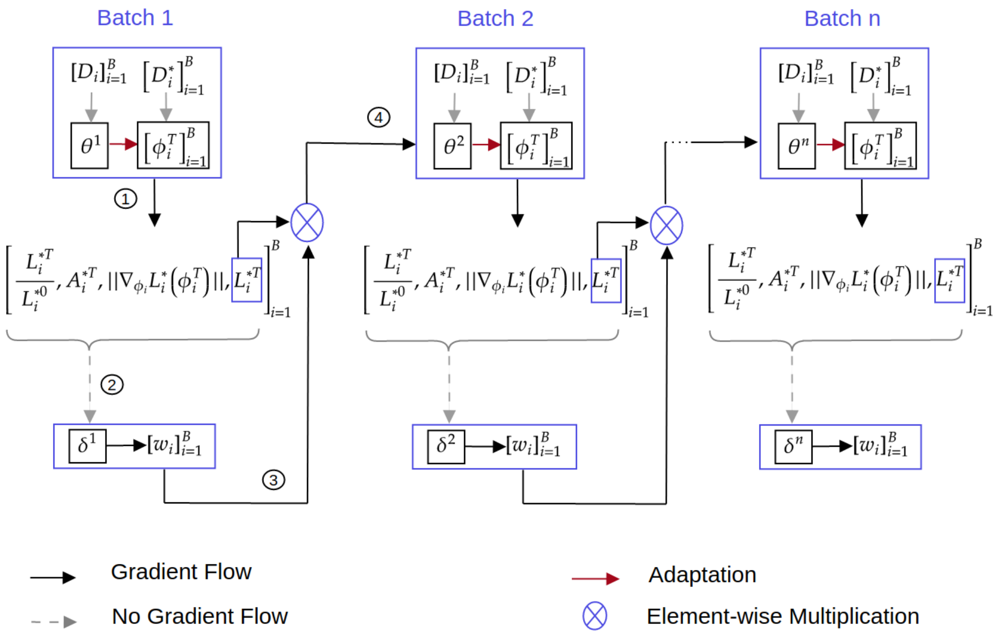

Figure 1: Computational Graph of the forward pass of the meta-model using task attended meta-training curriculum. The output of this procedure is a meta-model θ^n. Gradients are propagated through solid lines and restricted through dashed lines.

true. It is likely that given the current configuration of the meta-model, some tasks may be more important for the meta-model's learning. A contributing factor to this difference is that tasks sampled from complex data distributions can be profoundly diverse. The diversity and latent properties of the tasks coupled with the model configuration may induce some tasks to be better aligned with the optimal meta-knowledge than others. The challenging aspect in the meta-learning setting is to define the "importance" and associate weights to the tasks of a batch proportional to their contribution to improving the meta-knowledge. As human beings, we *learn* to associate importance to events subjective to meta-information about the events and prior experience. This motivates us to define a learnable module that can map the meta-information of tasks to their importance weights.

4.1 Characteristics of Meta-Information

Given a task-batch $\{\mathcal{T}_i\}_{i=1}^B$, the task attention module takes as input meta-information about each task ($\mathcal{T}_i$) in the batch, defined as the four tuple below:

$$\mathcal{I} = \left\{ \left(||\nabla_{\phi_i^T} L_i^*(\phi_i^T)||, L_i^{*T}, A_i^{*T}, \frac{L_i^{*T}}{L_i^{*0}} \right) \right\}_{i=1}^{B} \tag{1}$$

where corresponding to each task i in the batch $||\nabla_{\phi_i^T} L_i^*(\phi_i^T)||$ denotes the norm of gradient, L_i^{*T} and A_i^{*T} are the test loss and accuracy of the adapted model respectively, and $\frac{L_i^{*T}}{L_i^{*0}}$ is the ratio of the model's test loss post and prior adaptation.

4.1.1 Gradient norm

Let $P = \{\phi_i^T\}_{i=1}^B$ be the parameters of the models obtained after adapting the initial model (for T iterations) on the support data $\{D_i\}_{i=1}^B$ of tasks $\{\mathcal{T}_i\}_{i=1}^B$. Also, let $G = \left\{(\nabla_{\phi_i^T} L^*(\phi_i^T, D_i^*)\right\}_{i=1}^B$ be the gradients of the adapted model parameters w.r.t the query losses $\{L_i^*\}_{i=1}^B$. The gradient norm $\left\{||\nabla_{\phi_i^T} L_i^*(\phi_i^T)||\right\}_{i=1}^B$ is the L_2 norm of the gradients and quantifies the magnitude of the consolidated displacement of the adapted model parameters during a gradient descent update on query data. Larger gradient norm on query dataset could indicate that the model has either not learned the support set or has overfitted. Hence the model is not generalizable on query set compared to the models with low gradient norm. Gradient norm therefore carries information about the convergence and generalizability of the adapted models.

Algorithm 1: Task Attended Meta-Training

```
Input:
Dataset: $\mathcal{M} = \{D_i, D_i^*\}_{i=1}^M$
Models: Meta-model $\theta$, Base-model $\phi$, Att-module $\delta$
Learning-rates: $\alpha$, $\beta$, $\gamma$
Parameters: Iterations $n_{iter}$, Batch-size $B$,
            Adaptation-steps $T$
Output: Meta-model : $\theta$
Initialization: $\theta, \delta \leftarrow$ Random Initialization
for iteration in $n_{iter}$ do
    $\{\mathcal{T}_i\}_{i=1}^B = \{D_i, D_i^*\}_{i=1}^B \leftarrow$ Sample task-batch($\mathcal{M}$)
    for all $\mathcal{T}_i$ do
        $\phi_i^0 \leftarrow \theta$
        $L_i^{*0}, \_ \leftarrow evaluate(\phi_i^0, D_i^*)$
        $\phi_i^T = adapt(\phi_i^0, D_i)$
        $L_i^{*T}, A_i^{*T} \leftarrow evaluate(\phi_i^T, D_i^*)$
    end
    $[w_i]_{i=1}^B \leftarrow$
      $Att\_module\left(\left[\frac{L_i^{*T}}{L_i^{*0}}, A_i^{*T}, ||\nabla_{\phi_i^T} L_i^*(\phi_i^T)||, L_i^{*T}\right]_{i=1}^B\right)$
    $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{i=1}^B w_i L_i^*(\phi_i^T)$
    $\{D_j, D_j^*\}_{j=1}^B \leftarrow$ Sample task-batch($\mathcal{M}$)
    for all $\mathcal{T}_j$ do
        $\phi_j^0 \leftarrow \theta$
        $\phi_j^T = adapt(\phi_j^0, D_j)$
    end
    $\delta \leftarrow \delta - \gamma \nabla_\delta \sum_{j=1}^B L_j^*(\phi_j^T)$
end
Return $\theta$
```

4.1.2 Test Loss

$\{L_i^{*T}\}_{i=1}^B$ represents the empirical error (cross entropy loss) of the adapted base models on unseen query instances and hence characterizes their generalizability. Unlike grad norm, which characterizes the generalizability in parameter space, query loss quantifies generalizability in the output space as the divergence between the real and predicted probability distributions. As $\{L_i^{*T}\}_{i=1}^B$ is a key component in the meta-update equation, it is an important factor influencing the meta-model's learning. Further, test errors of classes have been widely used to determine their "easy or hardness" Bengio et al. (2009); Liu et al. (2021). Thus $\{L_i^{*T}\}_{i=1}^B$ acquaints the attention module with the generalizability aspect of task models and their influence in updating the meta-model.

4.1.3 Test Accuracy

$\{A_i^{*T}\}_{i=1}^B$ corresponds to the accuracies of $\{\phi_i^T\}_{i=1}^B$ on $\{D_i^*\}_{i=1}^B$ scaled in the range [0,1]. A_i^{*T} evaluates the thresholded predictions (predicted labels) unlike L_i^{*T}, which evaluates the confidence of the model's predictions on the true class labels. Two task models may predict the same class labels but differ in the confidence of the predictions. In such scenarios, neither loss nor accuracy is individually sufficient to comprehend this relationship among the tasks. So, the combination of these two entities is more reflective of the nature of the learned task models.

4.1.4 Loss-ratio

Let L_i^{*0} be the loss of θ on the D_i^*, and L_i^{*T} be the loss of the adapted model ϕ_i^T on D_i^*. The loss-ratio $\frac{L_i^{*T}}{L_i^{*0}}$ is representative of the relative progress of a meta-model on each task. Higher values (> 1) of the loss-ratio suggests adapting θ to D_i has an adverse effect on generalizing it to D_i^* (negative impact), while lower values (< 1) of the loss-ratio indicates the benefit of adaptation of θ on D_i (positive impact). Loss-ratio of exactly one signifies adaptation attributes to no additional benefit (neutral impact). Therefore, loss-ratio provides information regarding the impact of adaptation on each task for a given meta-model.

4.2 Task Attention Module

We learn a task attention module parameterized by δ, which attends to the tasks that contribute more to the model's learning i.e., the objective of the task attention module is to learn the relative importance of each task in the batch for the meta-model's learning. Thus the output of the module is a $B-$dimensional vector $\mathbf{w} = [w_1, \ldots, w_B]$, $(\sum_{i=1}^{B} w_i = 1)$ quantifying the attention-score (weight - w_i) for each task. The attention vector $\mathbf{w}$ is multiplied with the corresponding task losses of the adapted models $L_i^*(\phi_i^T)$ on the held-out datasets D_i^* to update the meta-model θ:

$$\theta^{t+1} \leftarrow \theta^t - \beta \nabla_{\theta^t} \sum_{i=1}^{B} w_i L_i^*(\phi_i^T) \tag{2}$$

After the meta-model is updated using the weighted task losses, we evaluate the goodness of the generated attention weights. We sample a new batch of tasks $\{D_j, D_j^*\}_{j=1}^B$ and adapt a base-model ϕ_j using the updated meta-model θ^{t+1} on the train data $\{D_j\}$ of each task. The mean test-loss of the adapted models $\{\phi_j^T\}_{j=1}^B$ reflect the goodness of the weights assigned by the attention-module in the previous iteration. The attention module δ is thus updated using the gradients flowing back into it w.r.t to this mean test-loss. The attention network is trained simultaneously with the meta-model in an end to end fashion using the update rule:

$$\delta^{t+1} \leftarrow \delta^t - \gamma \nabla_{\delta^t} \sum_{j=1}^{B} L_j^*(\phi_j^T) \tag{3}$$

where ϕ_j^T is adapted from θ^{t+1}.

4.3 Task Attended Meta-Training Algorithm

We demonstrate the meta-training curriculum using the proposed task attention in Figure 1, formally summarized in Algorithm 1. As with the classical meta-training process, we first sample a batch of tasks from the task distribution. For each task $\mathcal{T}_i$, we adapt the base-model ϕ_i using the train data $\{D_i\}_{i=1}^B$ for T time-steps (line 7). The meta-information about the adapted models for each task is then computed, comprising of the loss L_i^{*T}, the accuracy A_i^{*T}, the loss-ratio $\frac{L_i^{*T}}{L_i^{*0}}$ and gradient norm $||\nabla_{\phi_i^T} L_i^*(\phi_i^T)||$ on test data $\{D_i^*\}_{i=1}^B$. The meta-information corresponding to each task in a batch is given as input to the task attention module (Figure 1 - Label: ②) which outputs the attention vector (line 10). The attention vector in combination with test losses $\{L_i^*\}_{i=1}^B$ is used to update meta-model parameters θ (line 11, Figure 1 - Label: ④). We sample a new batch of tasks $\{D_j, D_j^*\}_{j=1}^B$ and adapt the base-models $\{\phi_j^T\}_{j=1}^B$ using the updated meta-model. We compute the mean test loss over the adapted base-models $\{L_j^*(\phi_j^T)\}_{j=1}^B$, which is then used to update the parameters of the task attention module δ (lines 12-17).

The attention network is designed as a stand-alone module to learn the mapping from the meta-information space to the importance of tasks in a batch. Thus, it is important to decouple the learning of the attention network from that of the meta-model. The parameters of the meta-model θ should not be directly dependent on that of the task attention module δ. We prevent it by enforcing $\nabla_\theta w_i L_i^*(\phi_i^T) = w_i \nabla_\theta L_i^*(\phi_i^T)$. Restricting the flow of gradients to the meta-model through the task attention module also enables us to evade the computational overhead generated by the product of gradients. Specifically, stopping the gradient flow frees us from computing $\nabla_{\delta^t}\theta^{t+1}.\nabla_{\phi^t}\delta^t.\nabla_{\theta^t}\phi^t$. The leading term $\nabla_{\delta^t}\theta^{t+1}$ in turn requires the computation of $\nabla_{\delta^t}.\nabla_{\theta^t}.\nabla_{\theta^t}$. Figure 1 demonstrates the paths along which gradient backflow is restricted and permitted as dashed and solid lines respectively.

5 Experiments and Results

We consider different few-shot learning settings on the benchmark datasets - miniImagenet, Fewshot Cifar 100 (FC100) and tieredImagenet to test the effectiveness of the proposed attention module. All the experimental results and comparisons correspond to our re-implementation of the ML algorithms integrated into learn2learn library Arnold et al. (2020) to ensure fairness and uniformity. We believe that integrating the proposed attention module and additional ML algorithms into the learn2learn library will benefit the ML community. We perform individual hyperparameter tuning for all the models over the same hyperparameter space to ensure a fair comparison. The architecture and hyper-parameter details are provided in the implementation details. The source code is publicly available.[1]

5.1 Datasets and Implementation Details

miniImagenet Vinyals et al. (2016) dataset comprises 600 color images of size 84 × 84 from each of 100 classes sampled from the Imagenet dataset. The 100 classes are split into 64, 16 and 20 classes for meta-training, meta-validation and meta-testing respectively. **Fewshot Cifar 100 (FC100)** Oreshkin et al. (2018) dataset has been created from Cifar 100 object classification dataset. It contains 600 color images of size 32 × 32 corresponding to each of 100 classes grouped into 20 superclasses. Among 100 classes, 60 classes belonging to 12 superclasses correspond to meta-train set, 20 classes from 4 superclasses to meta-validation set, and the rest to meta-test set. **tieredImagenet** Ren et al. (2018a) is a more challenging benchmark for few-shot image classification. It contains 779,165 color images sampled from 608 classes of Imagenet and are grouped into 34 super-classes. These super-classes are divided into 20, 6, and 8 disjoint sets for meta-training, meta-validation, and meta-testing. In line with the state-of-the-art literature Sun et al. (2020), we use miniImagenet, FC100 and tieredImagenet for evaluating the effectiveness of the proposed attention module as they are more challenging datasets comprising of highly diverse tasks.

Table 1: Comparison of few-shot classification performance of MAML and TA-MAML on miniImagenet dataset with meta-batch size 4 and 6 for 5 and 10-way (1 and 5-shot) settings. The ± represents the 95% confidence intervals over 300 tasks. Algorithms denoted by * are rerun on their optimal hyper-parameters. We observe that TA-MAML consistently performs better than MAML, and an increase in the tasks in a batch improves the performance of both MAML and TA-MAML. The hardware constraint restricts the study on a 10-way 5-shot setting (meta-batch size 6), and meta-batch size of 8 or higher (OOM is an acronym for Out of memory).

	Test Accuracy (%) on miniImagenet			
	5-Way		10-Way	
Model	1 Shot	5 Shot	1 Shot	5 Shot
	Batch Size 4			
MAML*	46.10 ± 0.19	60.16 ± 0.17	29.42 ± 0.11	41.98 ± 0.10
TA-MAML	**48.36 ± 0.23**	**62.48 ± 0.18**	**31.15± 0.11**	**43.70 ± 0.09**
	Batch Size 6			
MAML*	47.72 ± 1.041	63.45 ± 1.083	31.55 ± 0.626	OOM
TA-MAML	**49.14 ± 1.211**	**65.26 ± 0.956**	**32.62± 0.635**	OOM

We use the architecture from Finn et al. (2017) for the base model and a two-layer LSTM Ravi & Larochelle (2017) for the parametric optimizer. The task attention module is a ReLU activated neural network with a 1 × 1 convolutional layer followed by 2 fully connected layers with 32 neurons, and finally a softmax activation to generate the attention weights. We perform a grid search over 30 different configurations for 5000 iterations to find the optimal hyper-parameters for each setting. The search space is shared across all meta-training algorithms and datasets. The meta, base and attention model learning rates are sampled from a log uniform distribution in the ranges $\left[1e^{-4}, 1e-2\right]$, $\left[1e^{-2}, 5e^{-1}\right]$ and $\left[1e^{-4}, 1e^{-2}\right]$ respectively (see appendix for more details). The hyperparameter λ for TAML (Theil) is sampled from a log uniform distribution over the range of $\left[1e^{-2}, 1\right]$. The number of adaptation steps is fixed to 5 for all settings except for 10-way 5-shot setting, where we use 2 adaptation steps owing to the computational expenses. The meta-batch size is set to 4 for all settings Finn et al. (2017); Jamal & Qi (2019). However, we study its impact in Table 1. All models were trained for 55000 iterations (early stopping was employed for tieredImagenet) using the optimal set of hyper-parameters using an Adam optimizer Kingma & Ba (2015).

---

[1] https://github.com/taskattention/task-attended-metalearning.git

5.2 Influence of Task Attention on Meta-Training

As the task-attention (TA) is a standalone module, it can be integrated with any batch episodic training regimen. To facilitate this integration, we introduced a batch-wise training regimen for parametric optimizer (MetaLSTM++ Aimen et al. (2021)). The comparative analysis of MetaLSTM and MetaLSTM++ on miniImagenet, FC100, and tieredImagenet is presented in Table 2. It is evident from the results that batch-wise episodic training is effective than sequential episodic training.

Table 2: Comparison of few-shot classification performance of vanilla ML algorithms with their task attended versions on miniImagenet, FC100 and tieredImagenet datasets for 5 and 10-way (1 and 5-shot) settings. The ± represents the 95% confidence intervals over 300 tasks. Algorithms denoted by * are rerun on their optimal hyper-parameters for a fair comparison. Attention-based ML algorithms perform better than their corresponding vanilla approaches across all the settings. Further, MetaLSTM++ and TA-MAML perform better than MetaLSTM and TAML, respectively, across all settings and datasets.

	Test Accuracy (%)			
	5-Way		10-Way	
Model	1 Shot	5 Shot	1 Shot	5 Shot
	miniImagenet			
MAML*	46.10 ± 0.19	60.16 ± 0.17	29.42 ± 0.11	41.98 ± 0.10
TAML*	46.26 ± 0.21	53.40 ± 0.14	29.76 ± 0.11	36.88 ± 0.10
TA-MAML	**48.36 ± 0.23**	**62.48 ± 0.18**	**31.15± 0.11**	**43.70 ± 0.09**
MetaSGD*	47.65± 0.21	61.60 ± 0.17	30.09± 0.10	42.22 ± 0.11
TA-MetaSGD	**49.28 ± 0.20**	**63.37 ± 0.16**	**31.50± 0.11**	**44.06 ± 0.10**
MetaLSTM*	41.48 ± 1.02	58.87 ± 0.94	28.62 ± 0.64	44.03 ± 0.69
MetaLSTM++	48.00 ± 0.19	62.73 ± 0.17	31.16 ± 0.09	45.46 ± 0.10
TA-MetaLSTM++	**49.18 ± 0.17**	**64.89 ± 0.16**	**32.07± 0.11**	**46.66 ± 0.09**
ANIL*	46.92 ± 0.62	58.68 ± 0.54	28.84 ± 0.34	40.95 ± 0.32
TA-ANIL	**48.84 ± 0.62**	**60.80± 0.55**	**31.14± 0.34**	**42.52 ± 0.34**
	FC100			
MAML*	36.40 ± 0.38	46.76±0.21	23.93±0.14	31.14 ± 0.07
TAML*	38.00 ± 0.26	48.05± 0.13	21.60± 0.14	33.19± 0.07
TA-MAML	**39.86± 0.25**	**49.56 ± 0.13**	**25.46± 0.15**	**36.06± 0.08**
MetaSGD*	33.46 ± 0.23	43.96± 0.13	21.40±0.15	30.59± 0.07
TA-MetaSGD	**35.66±0.25**	**49.49± 0.12**	**23.80±0.15**	**32.08±0.07**
MetaLSTM*	37.20 ± 0.26	47.89 ± 0.13	21.70 ± 0.14	32.11 ± 0.07
MetaLSTM++	38.60 ±0.23	49.82 ± 0.12	22.80 ± 0.14	33.46 ± 0.08
TA-MetaLSTM++	**41.53 ±0.28**	**51.17 ±0.13**	**25.33 ±0.15**	**34.18 ±0.08**
ANIL*	34.08 ± 1.29	44.74 ± 0.68	20.65 ± 0.77	27.93 ± 0.42
TA-ANIL	**38.06 ± 1.26**	**46.94± 0.69**	**23.27± 0.79**	**28.29 ± 0.40**
	tieredImagenet			
MAML*	44.40 ± 0.49	57.07 ± 0.22	27.40 ± 0.25	34.30 ± 0.14
TAML*	46.40 ± 0.40	56.80 ± 0.23	26.40 ± 0.25	34.40 ± 0.15
TA-MAML	**48.40 ± 0.46**	**60.40 ± 0.25**	**31.00± 0.26**	**37.60± 0.15**
MetaSGD*	52.80 ± 0.44	62.35 ± 0.26	31.90 ± 0.27	44.16 ± 0.15
TA-MetaSGD	**56.20 ± 0.45**	**64.56 ± 0.24**	**33.20± 0.29**	**47.12 ± 0.16**
MetaLSTM*	37.00 ± 0.44	59.83 ± 0.25	29.80 ± 0.28	39.28 ± 0.13
MetaLSTM++	47.60 ± 0.49	63.24 ± 0.25	30.70 ± 0.27	47.97 ± 0.16
TA-MetaLSTM++	**49.00 ± 0.44**	**66.15 ± 0.23**	**32.10± 0.27**	**51.35 ± 0.17**
ANIL*	45.08 ± 1.37	59.71 ±0.77	29.32 ± 0.83	42.76 ± 0.50
TA-ANIL	**45.96 ± 1.32**	**60.96± 0.72**	**32.68± 0.92**	**47.56 ± 0.51**

We also investigate the performance of the models trained with the TA meta-training regimen with their non-TA counterparts. Specifically, we compare MAML, MetaSGD, MetaLSTM++ and ANIL with TA-MAML, TA-MetaSGD, TA-MetaLSTM++ and TA-ANIL respectively over 5 and 10-way (1 and 5-shot) settings on miniImagenet, FC100 and tieredImagenet datasets and report the results in Table 2. For ANIL and TA-ANIL, we consider 1000 testing tasks. We observe that models trained with TA regimen generalize better to the unseen meta-test tasks than their non-task-attended versions across all the settings in all datasets. Note that the proposed task attention mechanism aims not to surpass the state-of-the-art meta-learning algorithms but provides new insight into the batch episodic meta-training regimen, which as per our knowledge, is common to all meta-learning algorithms.

We also compare the performance of TA-MAML against TAML - a meta-training regimen that forces the meta-model to be equally close to all the tasks. The results, as presented in Table 2, suggest that TA-MAML performs better than TAML on all benchmarks across all settings.

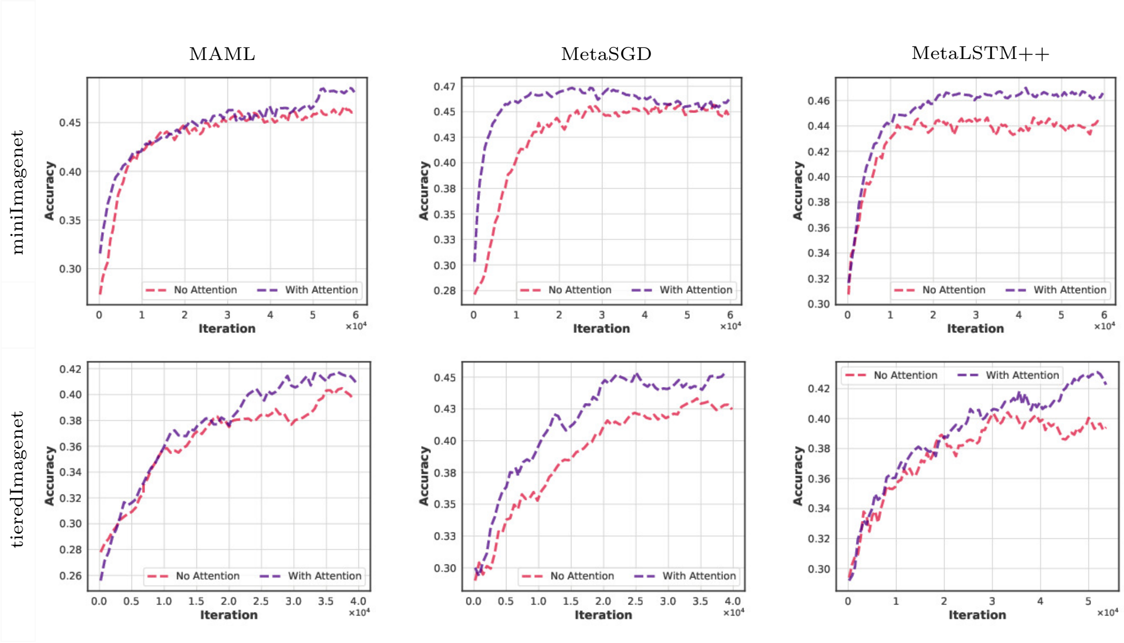

Figure 2: Mean validation accuracies of MAML (Col-1), MetaSGD (Col-2) and MetaLSTM++ (Col-3) across 300 tasks with/without attention on 5-way 1-shot setting on miniImagenet (Row-1) and tieredImagenet (Row-2) datasets.

Note that both TAML and TA-MAML are approaches that built upon MAML to address the inequality/diversity of tasks in a batch. Our aim is thus to compare TAML and TA-MAML and not to assess the efficacy of TAML when meta-trained using task attention. We investigate the influence of the TA meta-training regimen on the model's convergence by analyzing the trend of the model's validation accuracy over iterations. Figure 2 depicts the mean validation accuracy over 300 tasks on miniImagenet and tieredImagenet datasets for a 5-way 1-shot setting across training iterations. We observe that the models meta-trained with TA regimen tend to achieve higher/at-par performance in fewer iterations than the corresponding models meta-trained with the non-TA regimen.

Table 3: Effect of ablating components of meta-information in TA-MAML for 5 way 1 and 5 shot settings on miniImagenet dataset.

Ablation on inputs					
Grad norm	Loss	Loss-ratio	Accuracy	Test Accuracy	
				5-way 1-shot	5-way 5-shot
×	×	×	×	46.10±0.19	60.16±0.17
✓	✓	✓	×	47.30±0.16	60.48±0.16
✓	✓	×	✓	47.62±0.17	62.17±0.17
✓	×	✓	✓	48.10±0.18	60.90±0.20
×	✓	✓	✓	47.30±0.18	61.52±0.16
✓	✓	✓	✓	**48.36±0.23**	**62.48±0.18**

5.3 Ablation Studies

To examine the significance of each input given to the task attention model, we conduct an ablation study on 5-way 1 and 5 shot TA-MAML on miniImagenet dataset and report the results in Table 3. We observe that all the components of meta-information contribute to the learning of a more generalizable meta-model. To further support this observation, we analyze the ranks of the tasks for maximum and minimum values of : loss, loss ratio, accuracy, and grad norm in a batch, as per the weights across training iterations and the results are described in the supplementary material.

5.4 Relation of Weights with Meta-Information

We investigate the relationship between the meta-information and weights assigned by the task attention module by analyzing the mean Pearson correlation of each of the components (four tuple) of the meta-information with the attention vector across the training iterations. This is depicted in Figure 3 for TA-MAML on 5-way 1 and 5 shot settings for miniImagenet dataset. We observe that the loss ratio and loss are positively correlated with the attention vector, while accuracy and gradient norm is negatively correlated. In 5-way 5-shot setting, we observe that the correlation pattern is comparable to 5-way 1-shot setting, but the mean correlation value of grad norm across iterations is less than that of the 5-way 1-shot setting. This could be because the 5-way 5-shot setting is richer in data than the 5-way 1-shot setting, which allows better learning and therefore has low average values of grad norm (Section 4.1.1).

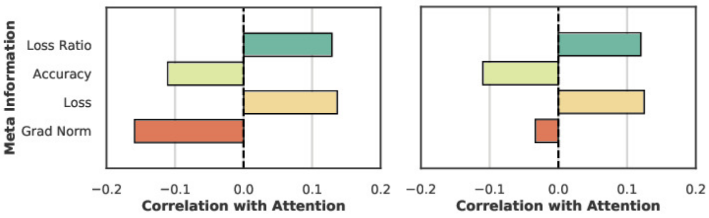

Figure 3: Mean Pearson correlation of TA-MAML on 5-way 1-shot (left) and 5-shot (right) setting on miniImagenet.

In figure 4, we illustrate the trend of mean weighted loss across iterations for TA-MAML on 5-way 1 and 5 shot settings on mini-Imagenet dataset. The trend indicates that the average weighted loss decreases over the meta-training iterations. The shaded region represents a 95% confidence interval over 100 tasks.

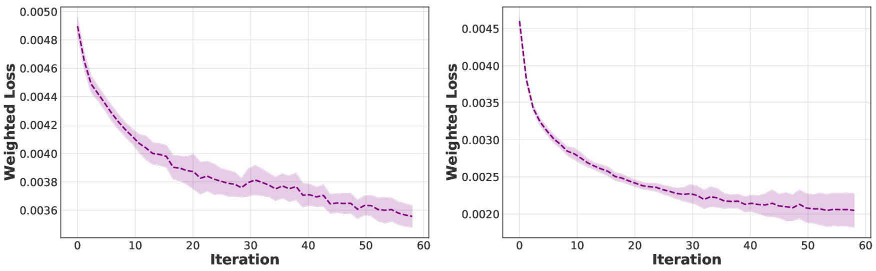

Figure 4: Trend analysis of weighted loss across meta-training iterations for TA-MAML on 5-way 1-shot (left) and 5-shot (right) settings on miniImagenet dataset. Iterations are in thousands.

5.5 Analysis of Attention Network

To gain further insights into the operation of the attention module, we also examine the trend of the attention-vector (Figure 5) while meta-training TA-MAML for 5 way 1 and 5 shot settings on the miniImagenet dataset. We plot the maximum and the minimum attention score assigned to the tasks of a batch across iterations together with a few weighted task batches in 5-way 1-shot setting for illustration. We note that the weighted task batches are only intended to demonstrate the change in the tasks' attention scores across iterations. The next experiment

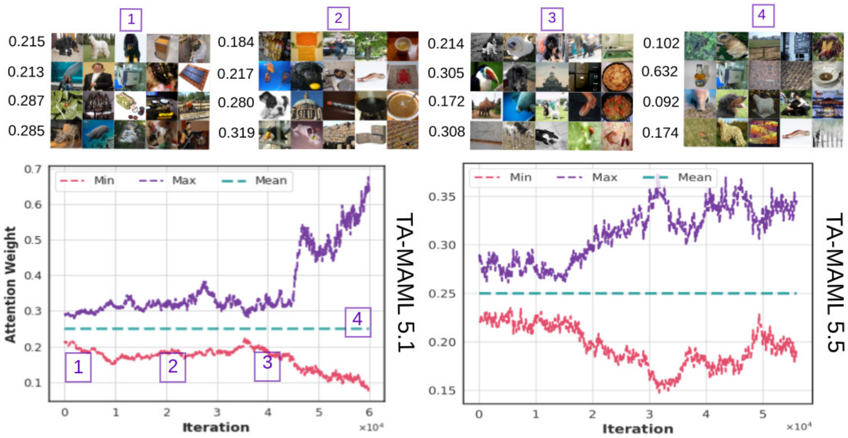

Figure 5: Trend of an attention vector in 5-way 1-shot (left) and 5-shot (right) settings on miniImagenet dataset for TA-MAML.

presents a more rigorous analysis studying the relationship among classes in a task and attention scores assigned. We also note that the mean attention score is always 0.25 as we follow a meta-batch size of 4. We observe that the TA module's output follows an interesting trend. Initially, the TA module assigns almost uniform weights to all the tasks of a batch; however, as the iterations increase, the TA module assigns unequal scores to the tasks in a batch, preferring some over the other. This suggests that during the initial phases of the meta-model's training, all tasks have equal contribution towards learning a *generic structure* of the meta-knowledge. As the meta-model's learning proceeds, learning the further *fine-grained meta-knowledge structure* requires prioritizing some tasks in a batch over the others, which are potentially better aligned with learning the optimal meta-knowledge. To reduce the computational burden, we freeze the weights of the attention module after 15000 iterations, i.e., only inputs of the attention module vary beyond 15000 iterations. We obtained a similar performance as when the attention module was trained throughout the meta-train phase (≈ 48% for 5-way 1-shot setting on miniImagenet dataset). From Figure 6, we observe that the attention vector still follows a similar trend as when trained end-to-end, indicating 15000 iterations are sufficient for the attention module's training. Thus, we note that proposed approach is computationally feasible. We further attempt to decipher the functioning of the black box attention network by analyzing the qualitative relation among weights and the classes of task batches (Figure 9 is presented in supplementary material due to space constraints). In Figure 9 left column (col-1) corresponds to the cases where the assignment of attention scores to the tasks is human interpretable. In contrast, the right column (col-2) refers to the uninterpretable attention scores. From the human perspective, tasks containing images from similar classes are hard to distinguish and are assigned higher attention scores indicated by red bounding boxes in Figure 9 (col-1). Specifically, (col-1, row-1) task 2 is regarded as most important, possibly because it includes three breeds of dogs followed by task 4, which comprises two species of fish. However, the aforementioned is not a hard constraint, as there are some task batches (Figure 9 (col-2)) in which the distribution of weights cannot be explained qualitatively.

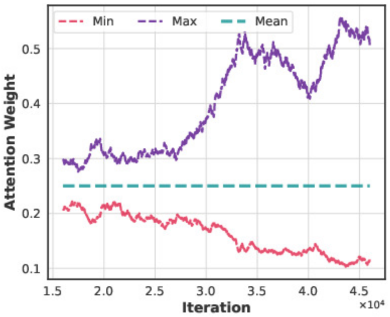

Figure 6: Trend of an attention vector for TA-MAML when attention module is frozen after 15000 iterations in 5-way 1-shot setting on miniImagenet dataset.

6 Conclusion

In this work we have shown that the batch wise episodic training regimen adopted by ML strategies can benefit from leveraging knowledge about the importance of tasks within a batch. Unlike prior approaches that assume uniform importance for each task in a batch, we propose task attention as a way to learn the relevance of each task according to its alignment with the optimal meta-knowledge. We have validated the effectiveness of task attention by augmenting it to popular initialization and parametric-optimization based ML strategies. To facilitate integration with the latter, we have introduced a batch wise training strategy for a parametric optimizer, that outperforms its previous sequential counterpart. We have demonstrated through few-shot learning experiments on miniImagenet, FC100 and tieredImagenet datasets that augmenting task attention helps attain better generalization to unseen tasks from the same distribution while requiring fewer iterations to converge. We also conduct an exhaustive empirical analysis on the distribution of attention weights to study the nature of the meta-knowledge and task attention module. We believe that this end-to-end attention-based meta training paves the way towards efficient and automated meta-training.

References

Aroof Aimen, Sahil Sidheekh, Vineet Madan, and Narayanan C Krishnan. Stress Testing of Meta-learning Approaches for Few-shot Learning. In *AAAI Workshop on Meta-Learning and MetaDL Challenge*, 2021.

Sébastien MR Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for meta-learning research. *CoRR*, 2020.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pp. 41–48. ACM, 2009.

Haw-Shiuan Chang, Erik G. Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1002–1012, 2017.

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

Ricardo Luna Gutierrez and Matteo Leonetti. Information-theoretic task selection for meta-reinforcement learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 11719–11727. Computer Vision Foundation / IEEE, 2019.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2309–2318. PMLR, 2018.

Jean Kaddour, Steindór Sæmundsson, and Marc Peter Deisenroth. Probabilistic active meta-learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

H. Kahn and A. W. Marshall. Methods of reducing sample size in monte carlo computations. *Oper. Res.*, 1953.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pp. 1189–1197. Curran Associates, Inc., 2010.

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

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 2999–3007. IEEE Computer Society, 2017.

Chenghao Liu, Zhihao Wang, Doyen Sahoo, Yuan Fang, Kun Zhang, and Steven C. H. Hoi. Adaptive task sampling for meta-learning. In *ECCV*, 2020.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *ICML*, 2021.

Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 719–729, 2018.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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 *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018a.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4331–4340. PMLR, 2018b.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 761–769. IEEE Computer Society, 2016.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 403–412. Computer Vision Foundation / IEEE, 2019.

Qianru Sun, Yaoyao Liu, Zhaozheng Chen, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning through hard tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3630–3638, 2016.

Huaxiu Yao, Yu Wang, Ying Wei, Peilin Zhao, Mehrdad Mahdavi, Defu Lian, and Chelsea Finn. Meta-learning with an adaptive task scheduler. *Advances in Neural Information Processing Systems*, 2021.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1–9. JMLR.org, 2015.

7 Appendix

7.1 Experiments

7.1.1 Ablation Studies

We analyze the ranks of the tasks for maximum and minimum values of : loss, loss ratio, accuracy, and grad norm in a batch wrt attention weights throughout meta-training of TA-MAML on a 5-way 1 and 5 shot settings on miniImagenet dataset (Figure 7 and 8). Specifically, the highest weighted task is given rank one, and the least weighted task in a batch is given the last rank. We observe that the TA module does not assign maximum weight to the tasks with maximum or minimum values of : test loss, loss ratio, grad norm or accuracy throughout meta-training. Thus, the TA module does not trivially learn to assign weights to the tasks based on some component of meta-information but learns useful latent information from all the components to assign importance for the tasks in a batch.

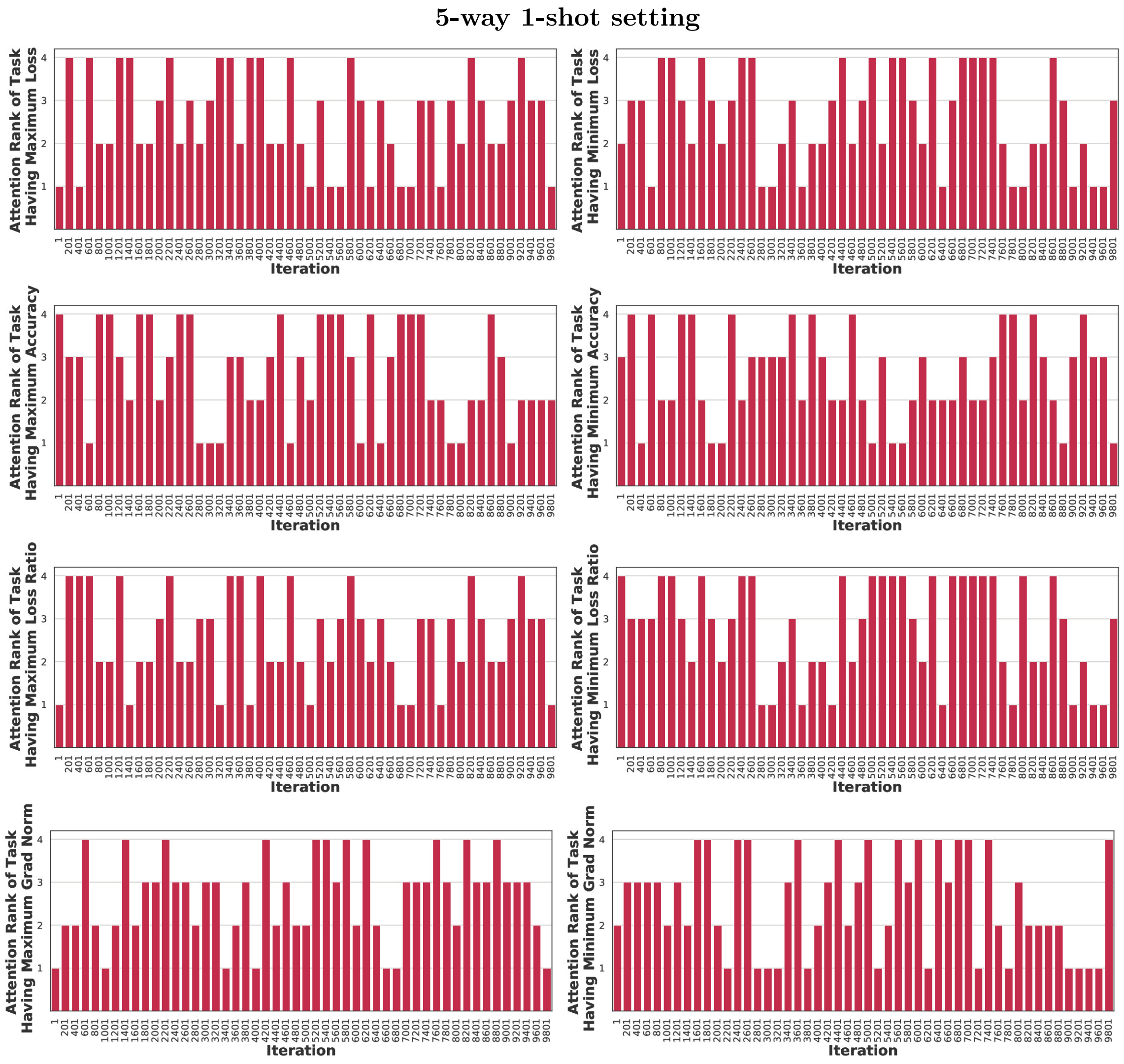

Figure 7: Rank Analysis of tasks for maximum and minimum values of : loss, loss-ratio, accuracy and grad norm throughout the training of TA-MAML for 5-way 1 shot setting on miniImagenet dataset.

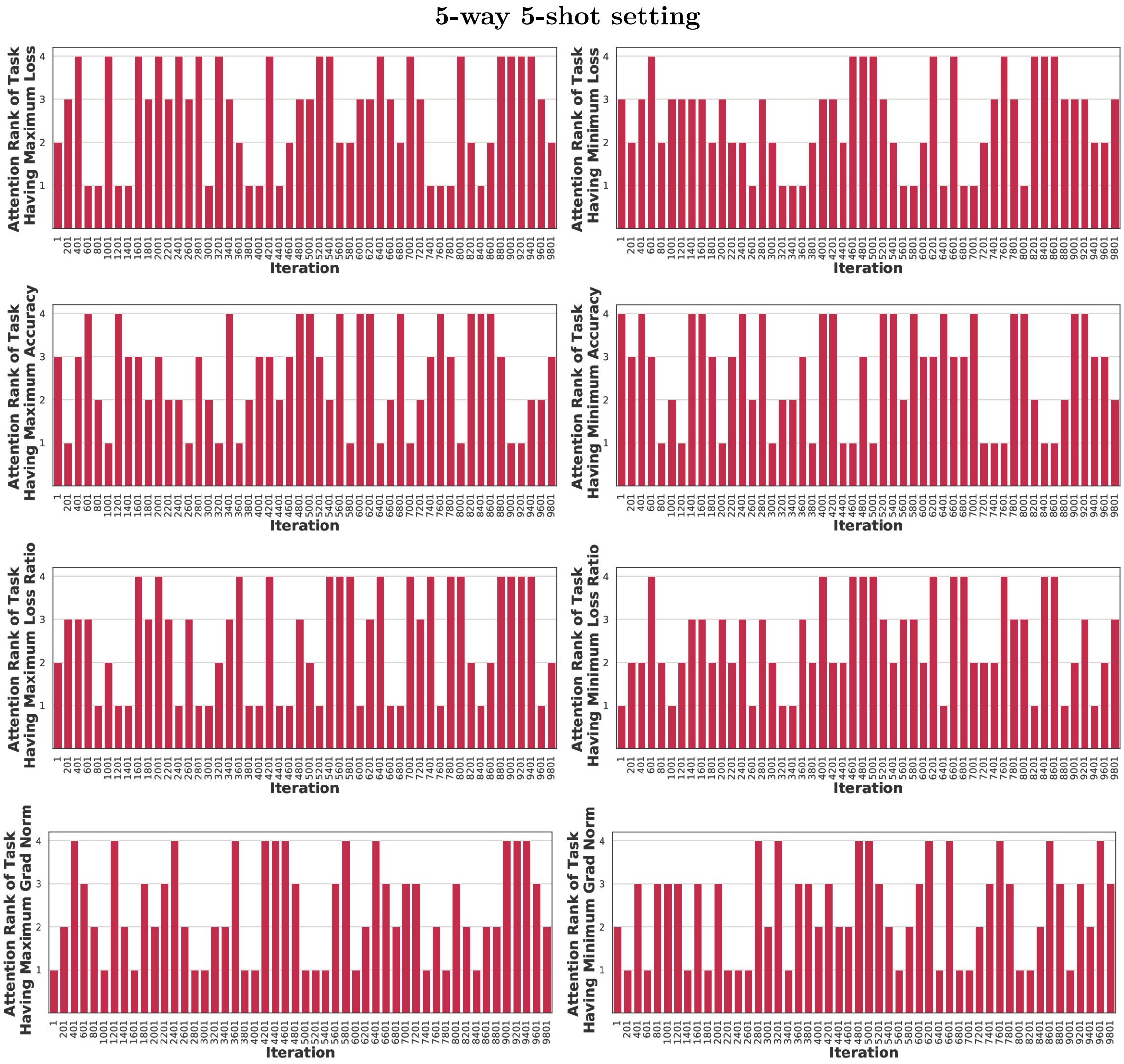

Figure 8: Rank Analysis of tasks for maximum and minimum values of : loss, loss-ratio, accuracy and grad norm throughout the training of TA-MAML for 5-way 5 shot setting on miniImagenet dataset.

7.1.2 Analysis of Attention Network

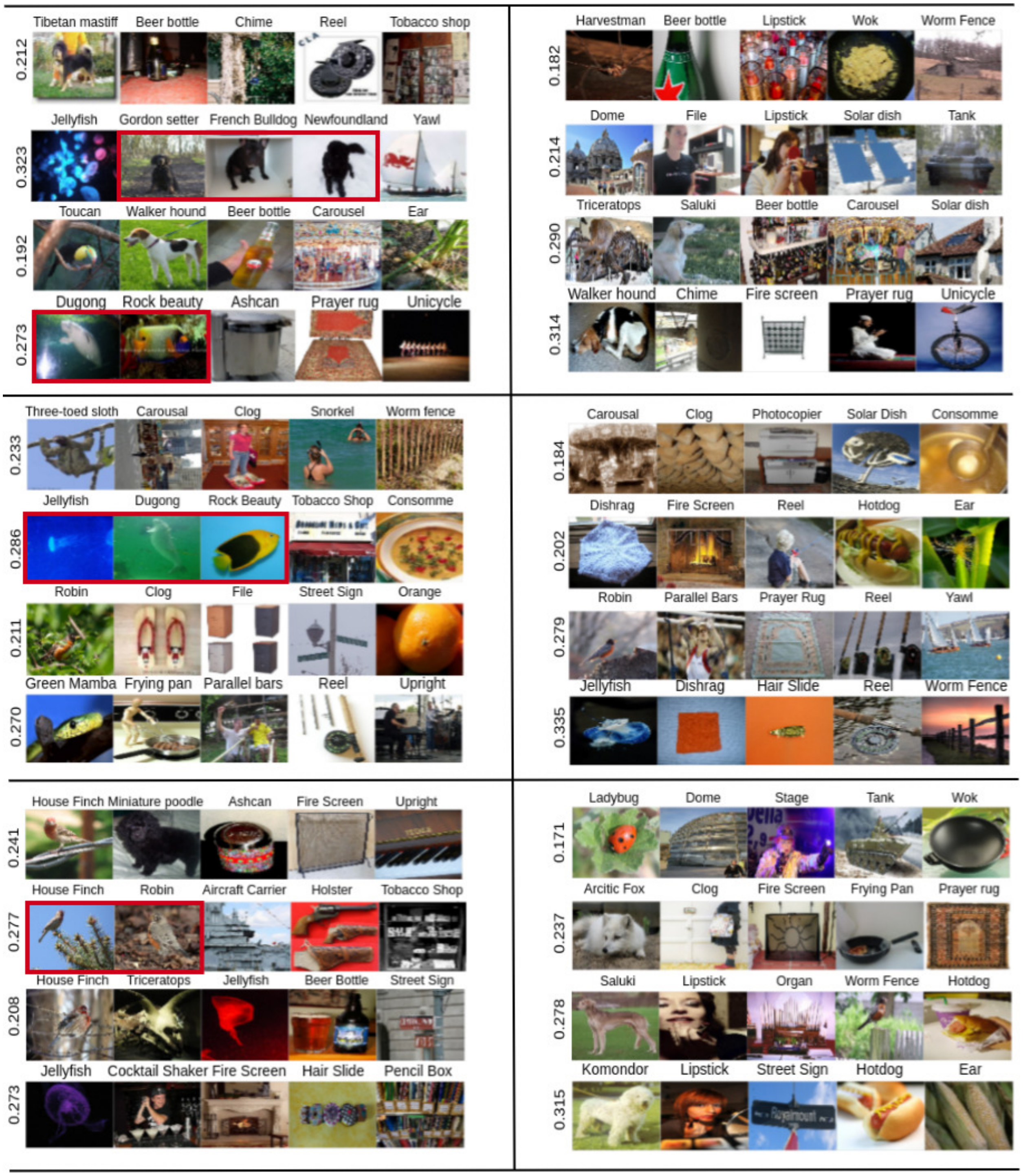

Figure 9: Explanations of TA module in TA-MAML on miniImagenet. **Left Col)** Higher weights accredited to tasks with comparable classes marked by red bounding boxes. **Right Col)** Association of weights and task data is qualitatively uninterpretable. Rows correspond to the batches.

7.1.3 Hyperparameter Details

Setting	Model	base lr	meta lr	attention lr	lambda
			miniImagenet		
5.1	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.0748
	TA-MAML	0.0763	0.0005	0.0004	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0529	0.0011	0.0004	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0012	-	-
	TA-MetaLSTM++	-	0.0012	0.0031	-
	ANIL	0.3000	0.0006	-	-
	TA-ANIL	0.0763	0.0005	0.0004	-
5.5	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.7916
	TA-MAML	0.0763	0.0005	0.0004	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0529	0.0011	0.0004	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0012	-	-
	TA-MetaLSTM++	-	0.0004	0.0001	-
	ANIL	0.3000	0.0006	-	-
	TA-ANIL	0.0763	0.0005	0.0004	-
10.1	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.2631
	TA-MAML	0.2551	0.0015	0.0001	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0627	0.0008	0.0013	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0015	-	-
	TA-MetaLSTM++	-	0.0009	0.0015	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.2551	0.0015	0.0001	-
10.5	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.0741
	TA-MAML	0.2551	0.0015	0.0001	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0627	0.0008	0.0013	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0036	-	-
	TA-MetaLSTM++	-	0.0024	0.0002	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.2551	0.0015	0.0001	-

Setting	Model	base lr	meta lr	attention lr	lambda
FC100					
5.1	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.0164
	TA-MAML	0.2826	0.0003	0.0024	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0349	0.0008	0.0001	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0010	-	-
	TA-MetaLSTM++	-	0.0002	0.0074	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.2826	0.0003	0.0024	-
5.5	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.0153
	TA-MAML	0.2826	0.0003	0.0024	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0349	0.0008	0.0001	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0002	-	-
	TA-MetaLSTM++	-	0.0007	0.0003	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.2826	0.0003	0.0024	-
10.1	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.0794
	TA-MAML	0.2353	0.0002	0.0001	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.2583	0.0029	0.0007	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0021	-	-
	TA-MetaLSTM++	-	0.0005	0.0014	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.2826	0.0003	0.0024	-
10.5	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.0193
	TA-MAML	0.2353	0.0002	0.0001	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.2583	0.0029	0.0007	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0004	-	-
	TA-MetaLSTM++	-	0.0004	0.0090	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.2826	0.0003	0.0024	-

Setting	Model	base lr	meta lr	attention lr	lambda
			tieredImagenet		
5.1	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.3978
	TA-MAML	0.02615	0.0005	0.0015	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0944	0.0003	0.0002	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0002	-	-
	TA-MetaLSTM++	-	0.0010	0.0006	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.0261	0.0005	0.0015	-
5.5	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.7733
	TA-MAML	0.0261	0.0005	0.0015	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0944	0.0003	0.0002	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0009	-	-
	TA-MetaLSTM++	-	0.0012	0.0001	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.0261	0.0005	0.0015	-
10.1	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.4752
	TA-MAML	0.0821	0.0002	0.0006	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0512	0.0007	0.0018	-
	MetaLSTM	-	0.005	-	-
	MetaLSTM++	-	0.0011	-	-
	TA-MetaLSTM++	-	0.0018	0.0002	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.0821	0.0002	0.0006	-
10.5	MAML	0.5000	0.0030	-	-
	TAML	0.5000	0.0030	-	0.2501
	TA-MAML	0.0821	0.0002	0.0006	-
	MetaSGD	0.5000	0.0030	-	-
	TA-MetaSGD	0.0512	0.0007	0.0018	-
	MetaLSTM	-	0.0050	-	-
	MetaLSTM++	-	0.0024	-	-
	TA-MetaLSTM++	-	0.0015	0.0019	-
	ANIL	0.5000	0.0030	-	-
	TA-ANIL	0.0821	0.0002	0.0006	-