TASK-AGNOSTIC ONLINE META-LEARNING IN NON-STATIONARY ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Online meta-learning has recently emerged as a marriage between batch meta-learning and online learning, for achieving the capability of quick adaptation on new tasks in a lifelong manner. However, most existing approaches focus on the restrictive setting where the distribution of the online tasks remains fixed with known task boundaries. In this work we relax these assumptions and propose a novel algorithm for task-agnostic online meta-learning in non-stationary environments. More specifically, we first propose two simple but effective detection mechanisms of task switches and distribution shift based on empirical observations, which serve as a key building block for more elegant online model updates in our algorithm: the task switch detection mechanism allows reusing of the best model available for the current task at hand, and the distribution shift detection mechanism differentiates the meta model update so as to preserve the knowledge for in-distribution tasks and quickly learn the new knowledge for out-of-distribution tasks. Motivated by the recent advance in online learning, our online meta model updates are based only on the current data, which eliminates the need of storing previous data as required in most existing methods. This crucial choice is also well supported by our theoretical analysis of dynamic regret in online meta-learning, where a sublinear regret can be achieved by updating the meta model at each round using the current data only. Empirical studies on three different benchmarks clearly demonstrate the significant advantage of our algorithm over related baseline approaches.

1 INTRODUCTION

Two key aspects of human intelligence are the abilities to quickly learn complex tasks and continually update their knowledge base for faster learning of future tasks. Meta-learning (Koch et al., 2015; Ravi & Larochelle, 2016; Finn et al., 2017) and online learning (Hannan, 1957; Shalev-Shwartz & Singer, 2007; Cesa-Bianchi & Lugosi, 2006) are two main research directions that try to equip learning agents with these abilities. In particular, meta-learning aims to facilitate quick learning of new unseen tasks by building a prior over model parameters based on the knowledge of related tasks, whereas online learning deals with the problem where the task data is sequentially revealed to a learning agent. To achieve the capability of fast adaptation on new tasks in a lifelong manner, online meta-learning (Finn et al., 2017; Harrison et al., 2020; Yao et al., 2020) has attracted much attention recently. Considering the setup where online tasks arrive one at a time, the objective of online meta-learning is to continuously update the meta prior based on which the new task can be learnt more quickly after the agent encounters more tasks.

In online meta-learning, the agent typically maintains two separate models, i.e., the *meta-model* to capture the underlying common knowledge across tasks and the *online task model* for solving the current task in hand. Most of the existing studies (Finn et al., 2017; Acar et al., 2021) in online meta-learning follow a "resetting" strategy: quickly adapt the online task model from the meta model using the current data, update the meta model and reset the online task model back to the updated meta model at the beginning of the next task. This strategy generally works well when the task boundaries are known and the task distribution remains stationary. However, in many real-world data streams the task boundaries are not directly visible to the agent (Rajasegaran et al., 2022; Caccia et al., 2020; Harrison et al., 2020), and the task distributions can dynamically change during the online learning stage. Therefore, in this work we seek to solve the online meta-learning problem in such more realistic settings.

Needless to say, how to efficiently solve the online meta-learning problem without knowing the task boundaries in the non-stationary environments is nontrivial due to the following key questions: (1) *How to update the meta model and the online task model?* Clearly, the "resetting" strategy at the moment of new data arriving is not desirable, as adapting from the previous task model is preferred when the new data belongs to the same task with the previous data. On the other hand, the meta model update should be distinct between in-distribution (IND) tasks, where the current knowledge should be preserved, and out-of-distribution tasks (OOD), where the new knowledge should be learnt quickly. (2) *How to make the system lightweight for fast online learning?* The nature of online meta-learning precludes sophisticated learning algorithms, as the agent should be able to quickly adapt to different tasks typically without access to the previous data. And dealing with the environment non-stationarity should not significantly increase the computational cost, considering that the environment could change fast during online learning.

The **main contribution** of this work is a **novel online meta-learning algorithm** in non-stationary environments without knowing the task boundaries, which appropriately addresses the problems above. More specifically, we first propose two simple but effective mechanisms to detect the task switches using the classification loss and detect the distribution shift using the Helmholtz free energy (Liu et al., 2020), respectively, as motivated by empirical observations. Based on these detection mechanisms, our algorithm provides a finer treatment on the online model updates, which brings in the following benefits: (1) (*task knowledge reuse*) The detection of task switches enables our algorithm to reuse the best model available for each task, avoiding the "resetting" to the meta model at each step as in most previous studies; (2) (*judicious meta model update*) The detection of distribution shift allows our algorithm to update the meta model in a way that the new knowledge can be quickly learnt for out-of-distribution tasks whereas the previous knowledge can be preserved for in-distribution tasks; (3) (*efficient memory usage*) Motivated by the advance in online learning (Mokhtari et al., 2016; Hazan et al., 2016) where updating the model online with the current data is sufficient to guarantee a sublinear regret, our algorithm does not reuse/store any of the previous data and updates the meta model at each online episode based only on the current data, which clearly differs from most existing studies (Finn et al., 2019; Yao et al., 2020; Rajasegaran et al., 2022) in online meta-learning. This design choice is also well supported by **our theoretical analysis** which shows that updating the meta model at each round with only the current data can lead to desirable sublinear *dynamic* regret growth. **Extensive experiments** in three different benchmarks clearly show that our algorithm significantly outperforms existing methods.

Related Work: Meta-learning. Also known as learning to learn, meta-learning (Finn et al., 2017; Vinyals et al., 2016; Li et al., 2017) is a powerful tool for leveraging past experience from related tasks to quickly learn good task-specific models for new unseen tasks. As a pioneering method that drives recent success in meta-learning, model-agnostic meta-learning (MAML) (Finn et al., 2017) seeks to find good meta-initialization such that one or a few gradient descent steps from the meta-initialization leads to a good task-specific model for a new task. Several variants of MAML have been introduced (Finn & Levine, 2018; Finn et al., 2018; Raghu et al., 2019; Rajeswaran et al., 2019; Nichol & Schulman, 2018; Nichol et al., 2018; Mi et al., 2019; Zhou et al., 2019). Other approaches are essentially model based (Santoro et al., 2016; Bertinetto et al., 2018; Ravi & Larochelle, 2016; Munkhdalai & Yu, 2017) and metric space based (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018).

Online Learning. In online learning (Hannan, 1957; Cesa-Bianchi & Lugosi, 2006; Hazan et al., 2007), the cost functions are sequentially revealed to an agent which is required to select an action before seeing each cost. One of the most studied approach is follow the leader (FTL) (Hannan, 1957), which updates the parameters at each step using all previously seen loss functions. Regularized versions of FTL have also been introduced to improve stability (Abernethy et al., 2009; Shalev-Shwartz et al., 2012). Similar in spirit to our work in terms of computational resources, online gradient descent (OGD) (Zinkevich, 2003) takes a gradient descent step at each round using only the revealed loss. However, traditional online learning methods do not efficiently leverage past experience and optimize for zero-shot performance without any adaptation. In this work, we study the online meta-learning problem, in which the goal is to optimize for quick adaptation on future tasks as the agent continually sees more tasks.

Continual Learning. Continual learning (CL; a.k.a lifelong learning) focuses on overcoming "catastrophic forgetting" (McCloskey & Cohen, 1989; Ratcliff, 1990) when learning from a sequence of non-stationary data distributions. Existing approaches are rehearsal-based (Lopez-Paz & Ranzato,

2017; Chaudhry et al., 2018; Riemer et al., 2018), regularization-based (Kirkpatrick et al., 2017; Aljundi et al., 2018), and expansion-based (Rusu et al., 2016; Yoon et al., 2018; Sarwar et al., 2019). For instance, rehearsal-based methods store a subset of previous tasks data and reuse it for experience "replay" to avoid forgetting. However, traditional CL methods evaluate the final model on all previously seen tasks so as to measure forgetting. In this work we are interested in online meta-learning and evaluate models with the average online performance (e.g., accuracy) after adaptation, which better captures the ability to quickly adapt to new online tasks (Caccia et al., 2020). Even though we do not specifically focus on avoiding forgetting, we update the meta-model in a way that preserves knowledge of in-distribution domain while also improving fast adaptation for out-of-distribution domains, as demonstrated in our various experiments.

Online Meta-learning. Online meta learning was first introduced in Finn et al. (2019). Pioneering methods (Finn et al., 2019; Yao et al., 2020; Yu et al., 2020) follow a FTL-like design approach, which requires storing previous tasks and leads to a linear growth of memory requirement. Follow-the-regularized-leader (FTRL) (Shalev-Shwartz et al., 2012) approach has also been extended to the online meta learning setting in (Balcan et al., 2019; Khodak et al., 2019), resulting in a better memory requirement. Acar et al. (2021) proposed a memory-efficient approach based on summarizing previous task experiences into one state vector. However, these approaches require knowledge of task boundaries and "reset" the task model to the meta model at each online episode (Finn et al., 2017; Denevi et al., 2019). Similar to Acar et al. (2021), our algorithm also overcomes the linear memory scaling. But unlike their method, our algorithm does not have access to task boundaries and can operate under dynamic environments. The method in Rajasegaran et al. (2022) tries to alleviate the "resetting" issue by updating the online model always starting from its previous state, which however needs to store previous models and has limited performance especially in dynamic environments where successive tasks can be very different.

None of the methods above considered the online meta-learning problem in a dynamic environment setting where the task distributions change substantially over time without knowing the task boundaries. Caccia et al. (2020) is the first work that empirically evaluated the proposed algorithm in a dynamic environment, but did not propose a method to quickly learn the knowledge for out-of-distribution tasks. In stark contrast, we update the meta representations in a way that preserves the in-distribution knowledge while continually improving fast adaptation for out-of-distribution tasks.

2 Background and Problem Formulation

Background. Before introducing the online meta-learning problem in non-stationary environments, we first briefly discuss some related concepts.

Meta-learning via MAML. Meta-learning (Finn et al., 2017; Vinyals et al., 2016; Li et al., 2017), a.k.a., learning to learn, seeks to quickly learn a new task with limited samples by leveraging the knowledge from similar tasks. More specifically, the objective therein is to learn a meta model based on a set of tasks $\{\mathcal{T}_i\}_{i=1}^M$ drawn from some unknown distribution $\mathbb{P}(\mathcal{T})$, from which task-specific models can be quickly obtained for new tasks from the same stationary distribution $\mathbb{P}(\mathcal{T})$. Taking MAML as an example, the objective therein is to learn a model initialization θ such that one or a few gradient descent steps from θ can lead to a good model for a new task $\mathcal{T} \sim \mathbb{P}(\mathcal{T})$, by solving the following optimization problem with training tasks $\{\mathcal{T}_i\}_{i=1}^M$:

$$\theta := \arg\min_{\theta} \frac{1}{M} \sum\nolimits_{i=1}^{M} f_i\left(U_i(\theta)\right) \tag{1}$$

where the task model $\phi_i = U_i(\theta) = \theta - \alpha \nabla \hat{f}_i(\theta)$, $\hat{f}_i$ and f_i correspond to the training and test losses for task $\mathcal{T}_i$, respectively.

Online learning. In the general online learning problem, loss functions are sequentially revealed to a learning agent: at each step t, the agent first selects an action θ_t, and then a cost $f_t(\theta_t)$ is incurred. The goal of the agent is to select a sequence of actions so as to minimize the following *static* regret

$$\mathrm{R}(T) = \sum\nolimits_{t=1}^{T} f_t\left(\theta_t\right) - \min_{\theta} \sum\nolimits_{t=1}^{T} f_t\left(\theta\right), \tag{2}$$

i.e., the gap between the agent's predictions $\{f_t(\theta_t)\}_{t=1}^T$ and the performance of the best *static* model in hindsight. A successful agent achieves a regret $\mathrm{R}(T)$ that grows sublinearly in T. Online learning is a well studied field and we refer the interested readers to Hazan et al. (2016) for more information.

Online meta-learning. As a marriage between online learning and meta-learning, online meta-learning (Finn et al., 2019; Yao et al., 2020; Harrison et al., 2020) aims to achieve the following two features: (i) fast adaptation to the current task (the meta-learning aspect); (ii) learn to adapt even faster as it sees more tasks (the online learning aspect). Specifically, the agent observes a stream of tasks $\mathcal{S} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$ sampled from $\mathbb{P}(\mathcal{T})$, where tasks are revealed one at a time. For each task $\mathcal{T}_t$, the agent has access to a support set S_t for task-specific adaptation and a query set Q_t for evaluation. The goal here is to select a sequence of meta models $\{\mathbf{w}_t\}$ for achieving sublinear growth of the following regret

$$\mathrm{R}_{\mathrm{meta}}(T) = \sum\nolimits_{t=1}^{T} f_t\left(\boldsymbol{U}_t\left(\theta_t\right)\right) - \min_{\theta} \sum\nolimits_{t=1}^{T} f_t\left(\boldsymbol{U}_t(\theta)\right) \tag{3}$$

where $\boldsymbol{U}_t$ is the task adaptation function depending on the support set S_t, and the cost function f_t is evaluated using the adapted parameters $\boldsymbol{U}_t\left(\mathbf{w}_t\right)$ on the query set Q_t. Intuitively, the agent seeks to learn a better meta model which leads to better task models for future tasks after seeing more tasks.

Online meta-learning in non-stationary environments. Differently from most online meta-learning studies (Finn et al., 2019; Yao et al., 2020; Yu et al., 2020; Harrison et al., 2020; Rajasegaran et al., 2022; Acar et al., 2021), in this work we consider the online meta-learning problem in a more realistic scenario:

Pre-trained meta model. In many real applications, there is plenty of data available for pre-training, and it is unrealistic to deploy an agent in complex dynamic environments without any basic knowledge of the tasks at hand (Caccia et al., 2020). Therefore, following the same line as in Caccia et al. (2020), we assume that there is a set of training tasks $\{\mathcal{T}_i^0\}_{i=1}^M$ drawn from some unknown distribution $\mathbb{P}^0(\mathcal{T})$. And as standard in meta-learning, each pre-training task $\mathcal{T}_i^0$ has a support dataset S_i^0 and a query dataset Q_i^0. In this work, we employ MAML over the training tasks to learn a pre-trained meta model.

Unknown task boundaries. During the online meta-learning phase, we assume that the task boundaries are unknown, i.e., the so-called task-agnostic setup (Caccia et al., 2020), in the sense that the agent does not know if the new coming data at time t belongs to the previous task or a new task. To model the uncertainty about task boundaries, we assume that at any time t the new data belongs to the previous task with probability $p \in (0, 1)$ or to a new task with probability $1 - p$.

Non-stationary task distributions. During the online meta-learning phase, the agent could encounter new tasks that are sampled from other distributions instead of the pre-training one $\mathbb{P}^0(\mathcal{T})$. To capture this non-stationarity in task distribution, we assume that whenever a new task arrives during online learning, it will be sampled either from $\mathbb{P}^0(\mathcal{T})$ with probability $\eta \in (0, 1)$ or from a new (w.r.t. $\mathbb{P}^0(\mathcal{T})$) distribution with probability $1 - \eta$. Note that we do not restrict the number of new distributions that can be encountered during online learning and the task distributions can be revisited.

3 Proposed Algorithm under distribution shifts

To address the online meta-learning problem mentioned above for non-stationary environments, we next propose a simple but effective algorithm, called onLine mEta lEarning under Distribution Shifts (LEEDS), based on the detection of task switches and distribution shift to assist fast online learning. Following most studies (Rajasegaran et al., 2022; Caccia et al., 2020) in online meta-learning, we maintain two separate models during the online learning stage: θ for the meta model and ϕ for the online task model.

Detection of task switches and distribution shift: To enable fast learning of a new task in online learning, it is clear that the detection mechanisms can not be overly sophisticated, but in the meanwhile have to be efficient with high detection accuracy. Towards this end, we propose two different methods for detecting the task switch and the distribution shift, respectively, which work in concert as key components of LEEDS.

Detection of task switches. To understand the learning behaviors under task switches, we evaluate the classification loss of the previous task model using the newly coming data, i.e., $\mathcal{L}(\phi_{t-1}; S_t)$ for time t, where $\mathcal{L}$ is the loss function, ϕ_{t-1} is the previous online model at time $t-1$, and S_t is the current support set. The left plot in Fig. 1 shows the empirical results on an online few-shot image recognition problem. As depicted, the loss value keeps decreasing as the agent receives more data from the same task but suddenly increases whenever a new task arrives. This is clearly reasonable as the learnt online model for the previous task does not fit the new task anymore. Inspired by this

empirical observation, we use a simple mechanism based on the value of $\mathcal{L}(\phi_{t-1}; S_t)$ to detect the task boundaries: there is a task switch whenever the loss is above some pre-defined threshold. As demonstrated later in Section 5, such a simple mechanism is indeed quite effective as corroborated by its high detection accuracies on various online meta-learning problems.

Detection of distribution shift. To efficiently determine if a new task is IND or OOD, i.e., sampled from the pre-training task distribution or not, we consider an energy-based OOD detection mechanism with a binary classifier $C_\tau(\cdot; \theta)$ defined as follows

$$C_\tau(\mathbf{x}; \theta) = \begin{cases} 1 & \text{if } -\mathrm{E}(\mathbf{x}; \theta) \leq \tau \\ 0 & \text{if } -\mathrm{E}(\mathbf{x}; \theta) > \tau \end{cases} \tag{4}$$

where $\mathrm{E}(\mathbf{x}; \theta) = -\delta \log \sum_{k=1}^{K} \exp\left(-g_k(\mathbf{x}; \theta)/\delta\right)$ corresponds to the Helmholtz free energy for input $\mathbf{x}$, $g_k(\cdot; \theta)$ is the k-th component of the prediction model's output, and δ is the temperature parameter. The hyperparameter τ is a threshold that can be set using the pre-training tasks. As shown in Liu et al. (2020), the negative energy is linearly aligned with the likelihood of the of input $\mathbf{x}$ under model θ, making it a useful score for distinguishing IND and OOD tasks.

Method	Memory
Ours	$\mathcal{O}(1)$
Rajasegaran et al. (2022)	$\mathcal{O}(T)$
Caccia et al. (2020)	$\mathcal{O}(1/p)$
Acar et al. (2021)	$\mathcal{O}(1)$
Finn et al. (2019)	$\mathcal{O}(T)$

Figure 1: **Left plot:** Variations of the online loss for a pre-trained meta model using MAML which is deployed for online learning. Red dot at 0 means no task switch at that time, and at 1 means the task switched at that time. **Right table:** Comparison of the memory requirements among different methods. T is number of online rounds and $p \in (0, 1)$ is non-stationarity level.

Update of meta and online parameters: Based on the two detection schemes, the next question is how to update the meta and task models accordingly for enabling fast adaption in dynamic environments.

Without such detection mechanisms, previous online meta-learning algorithms (Finn et al., 2019; Acar et al., 2021) typically adapt the task model from the meta model using a support set, evaluate the adapted model on a query set, and reset the task model to the meta model when new data is received from the online data stream process, and then repeat the process again. However, such a "resetting" scheme can be sub-optimal in realistic scenarios. For instance, if the newly received data belongs to the same task with the previous data, the agent should update the task model by starting from previously adapted parameters instead of from the meta model.

In contrast, the simple but effective detection mechanisms in this work enable a more elegant treatment to the knowledge update during online learning:

(1) If there is a task switch at time t, i.e., $\mathcal{L}(\phi_{t-1}; S_t) > \ell$ where ℓ is the threshold, adapting from the meta model is generally better than adapting from the task model of the previous task. Therefore, we first obtain the online task model ϕ_t from the meta model using the new data:

$$\phi_t = \theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_{\theta_{t-1}} \mathcal{L}(\theta; S_t),$$

and then update the meta model no matter if there is a distribution shift, so as to incorporate the knowledge of the new task to the meta model:

$$\theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}}; Q_t).$$

(2) If there is no task switch, i.e., $\mathcal{L}(\phi_{t-1}; S_t) \leq \ell$, we continue to update the task model from the previous task model using the new data, different from the "resetting" scheme in the literature:

$$\phi_t = \phi_{t-1} - \alpha_1 \nabla_{\phi_{t-1}} \mathcal{L}(\phi_{t-1}; S_t).$$

Algorithm 1 onLine mEta lEarning under Distribution Shifts (LEEDS)

Input: Dynamic stream $\mathcal{S}$, pre-training distribution $\mathbb{P}^0(\mathcal{T})$, stepsizes α_1 and α_2, thresholds ℓ and τ.
Perform pre-training phase using MAML method on tasks drawn from $\mathbb{P}^0(\mathcal{T})$
while stream $\mathcal{S}$ is ON **do**
  $D_t \longleftarrow \mathcal{S}$ // receive current data from online data stream S_t, Q_t
  $S_t, Q_t \longleftarrow D_t$ // split data into support and query
  if $\mathcal{L}(\phi_{t-1}; S_t) \leq \ell$ (i.e., no switch) **then**
    $\phi_t = \phi_{t-1} - \alpha_1 \nabla_{\phi_{t-1}} \mathcal{L}(\phi_{t-1}; S_t)$ // adapt starting from previous online model
    Evaluate ϕ_t on query set Q_t
    if $C_\tau(S_t; \theta_{t-1})$ (i.e., covariate shift) **then**
      $\theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_\theta \mathcal{L}(\theta_{t-1}; S_t)$
      $\theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}}; Q_t)$ // update meta model
    end if
  else
    $\theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_{\theta_{t-1}} \mathcal{L}(\theta_{t-1}; S_t)$ // adapt starting from meta model using support set
    Set $\phi_t = \theta_{\text{adapt}}$ and Evaluate ϕ_t on query set Q_t
    $\theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}}; Q_t)$ // update meta model
  end if
end while

To accelerate the knowledge learning for the new domains, we further distinguish the meta model update for IND and OOD tasks. In particular, if the current task is an IND task, we will only update the meta model once at the beginning of this task. That is to say, the meta model will not be further updated within the same task. In stark contrast, if the current task is an OOD task, we continue to update the meta model whenever new data for this task arrives as follows:

$$\theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_\theta \mathcal{L}(\theta_{t-1}; S_t),\ \theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}}; Q_t).$$

Memory friendly: One important feature of the proposed algorithm LEEDS is that the meta model update is only based on the current data, as motivated by the theoretical advance in online learning (Mokhtari et al., 2016; Hazan et al., 2016) where updating the online model with only the current data is sufficient to guarantee a sublinear regret. This design is also well supported by our theoretical analysis in the following section, which indicates that updating the meta model with the current data is also sufficient for online meta-learning to achieve a *dynamic* regret that grows sublinearly. As a comparison, most of the previous studies store the previous data in the memory for the meta model update. A comparison of the memory requirements among different approaches is summarized in the right table in Fig. 1.

4 Theoretical Results

In the online meta-learning with distribution shifts, it is clear that the *static* comparator in eq. (3) is not sufficient to capture the non-stationarity, as one cannot expect to have a single meta-model for all task distributions. Hence, we consider the following more sophisticated *dynamic* regret (Mokhtari et al., 2016; Hazan et al., 2016; Zinkevich, 2003)

$$\mathrm{R}_{\text{meta}}(\mathbf{z}_1, ..., \mathbf{z}_T) = \sum\nolimits_{t=1}^{T} f_t\left(\boldsymbol{U}_t\left(\theta_t\right)\right) - \sum\nolimits_{t=1}^{T} f_t\left(\boldsymbol{U}_t(\mathbf{z_t})\right), \tag{5}$$

where the meta comparator $\mathbf{z}_t$ can change over time. If $\mathbf{z}_t = \arg\min_\theta f_t\left(\boldsymbol{U}_t\left(\theta\right)\right)$, then eq. (5) becomes the worst-case dynamic regret formulation. To make the dynamic regret in eq. (5) more suited to the realistic scenario of online meta-learning in which the task distribution changes after some time-steps, we consider the setting where the d-th encountered distribution stays stationary for K_d tasks before the distribution shift, leading to the following dynamic regret:

$$\begin{aligned}\mathrm{R}_{\text{meta}}(\theta_1^*, ..., \theta_D^*) &= \sum_{d=1}^{D}\sum_{k=1}^{K_d} f_d^k\left(\boldsymbol{U}_d^k\left(\theta_d^k\right)\right) - \sum_{d=1}^{D}\min_\theta \sum_{k=1}^{K_d} f_d^k\left(\boldsymbol{U}_d^k(\theta)\right)\\ &= \sum_{d=1}^{D}\sum_{k=1}^{K_d} f_d^k\left(\boldsymbol{U}_d^k\left(\theta_d^k\right)\right) - \sum_{d=1}^{D}\sum_{k=1}^{K_d} f_d^k\left(\boldsymbol{U}_d^k(\theta_d^*)\right)\end{aligned} \tag{6}$$

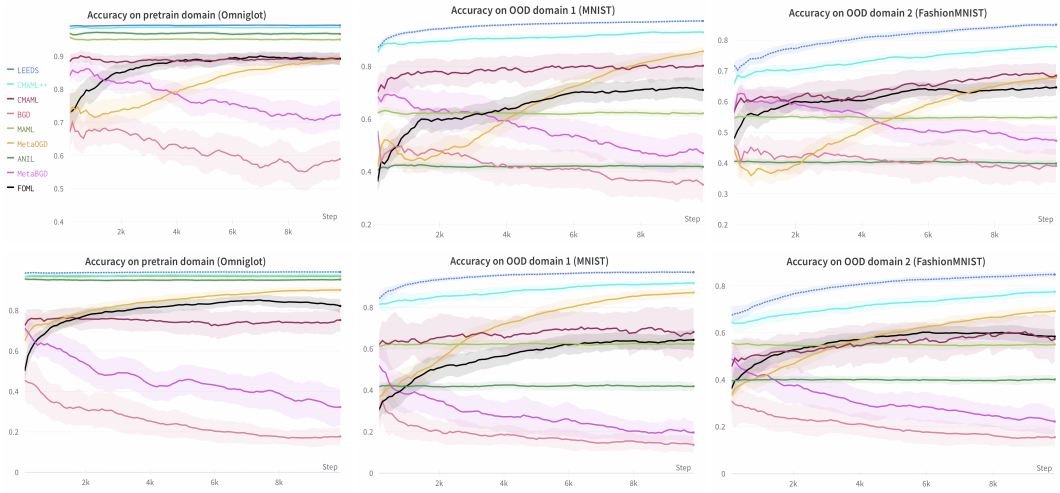


Figure 2: Online evaluations in each of the encountered domains during online learning phase for the **Omniglot-MNIST-FashionMNIST** benchmark. First row corresponds to non-stationarity level $p = 0.9$. In second row $p = 0.75$. LEEDS is the only method that is able to preserve pre-training knowledge while substantially increasing performance in OOD domains. Legend in first plot only.

where the meta comparator $\theta_d^* = \arg\min_\theta \sum_{k=1}^{K_d} f_d^k \left(\boldsymbol{U}_d^k(\theta)\right)$. Note that the task distributions can be revisited, i.e., the i-th distribution could be the same as the $(i+j)$-th distribution but distinct to the $(i+j-1)$-th for $j > 1$.

To show that updating the meta model based only on the current data (which is consistent with our algorithm) is sufficient for achieving sublinear regret, we consider the following update rule which updates the meta model at each step based on the current task only without reusing any previous data:

$$\theta_d^{k+1} := \theta_d^k - \alpha_d^k \nabla f_d^k \left(U_d^k(\theta_d^k)\right), \quad \text{with } \theta_{d+1}^1 := \theta_d^{K_d+1}. \tag{7}$$

And we make the following standard assumptions:

Assumption 1. *Each function f_d^k has bounded gradient norm, i.e., $\left\|\nabla f_d^k(\mathbf{w})\right\| \leq G \; \forall \; \mathbf{w}$.*

Assumption 2. *Each composition function $f_d^k \circ U_d^k$ is convex.*

We can have the following theorem which shows that updating the meta model based only on the current task without storing any previously seen data can achieve sublinear growth of the regret formulation in 6.

Theorem 1. *Suppose that Assumptions 1 and 2 hold. Let $P_D = \sum_{d=2}^{D} \|\theta_d^* - \theta_{d-1}^*\| + 1$ and $\alpha_d^k = \frac{1}{\sqrt{\sum_{d=1}^{D} K_d}}$. Then the dynamic regret is bounded as $\mathrm{R}_{\mathrm{meta}}(\theta_1^*, ..., \theta_D^*) \leq \mathcal{O}\left(P_D \sqrt{\sum_{d=1}^{D} K_d}\right)$.*

Further, if $\alpha_d^k = \sqrt{\frac{P_D}{\sum_{d=1}^{D} K_d}}$, then $\mathrm{R}_{\mathrm{meta}}(\theta_1^*, ..., \theta_D^*) \leq \mathcal{O}\left(\sqrt{P_D \sum_{d=1}^{D} K_d}\right)$.

We have the following remarks on Theorem 1. (a) Theorem 1 shows that under Assumptions 1 and 2, the dynamic regret 6 associated with the algorithm defined in eq. (7) grows sublinearly with respect to the total number of steps $\sum_{d=1}^{D} K_d$. This implies that the gap between the average loss (per task) of online models and that of optimal models in hindsight is asymptotically decreasing as more tasks are observed. (b) The dynamic regret depends on P_D (see its definition in the statement of Theorem 1), which captures the accumulative model shift (that further reflects the domain distribution shift over time). Clearly, smaller shift implies lower regret over time. (c) The dependency on P_D also shows that the upper bound becomes automatically tighter when the model shift across domains becomes small (i.e., dynamic comparators are close to each other), and thus the upper bound is adaptive. (d) Comparing the two regret bounds obtained in Theorem 1, we note that the dependency of the dynamic regret on P_D can be improved by appropriately selecting α_d^k when P_D is known in advance.

5 EXPERIMENTS

We conduct extensive experiments to answer the following questions: (i) How our algorithm perform when compared to existing online meta-learning methods for dynamic and static environments? (ii) How certain characteristic of the environment such as the non-stationarity level affects the performance of the algorithms? How our detection schemes perform when deployed to realistic dynamic environments. Further, we conduct ablation studies to investigate the advantage of the key domain adaptation module that permits our algorithm to process differently IND and OOD tasks. More comprehensive experimental results are provided in Appendix B due to space limitation.

Experimental setup. We pre-train the meta model in one domain and then deploy it in a dynamic environment where tasks can be drawn from new domains. We evaluate all the algorithms using the average of test losses obtained throughout the entire online learning stage. To investigate the impact of the non-stationary level on the learning performance, we further consider two different cases of the environment non-stationarity: A moderately stationary case where the probability of not switching to a new task is set to $p = 0.9$, and a low stationary case where $p = 0.75$. We do not consider the cases where p is very small, as an algorithm that just assumes task switches at each round should perform well in such cases. We compare algorithms over 10000 episodes unless otherwise stated. Due to space limitation, we defer details about datasets and baselines in Appendix C.

For all the experiments, whenever a new task needs to be revealed, it will be drawn from either the pre-training domain with probability 0.5, or from one of the OOD domains with probability 0.5. For the Tiered-ImageNet dataset, because only ood2 is trully OOD with respect to the pre-training task distribution, we increase the sampling probability of ood2 to 0.5 which is consistent to the protocol $50\% - 50\%$ for IND and OOD tasks in all our experiments. More details about the experimental setup including the neural network architectures and the hyperparameter search are deferred to Appendix D.

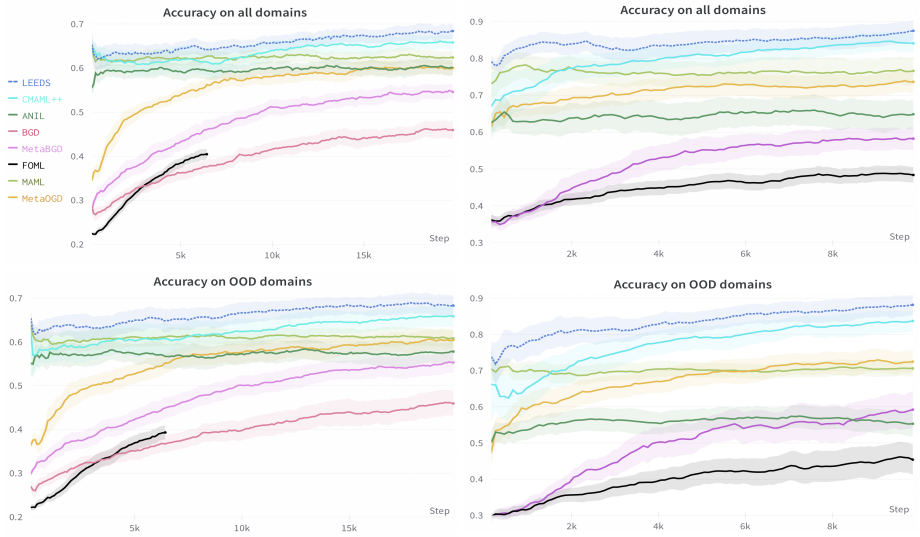


Figure 3: Online evaluations for the **Tiered-ImageNet (TI)** and **Synbols (SB)** benchmarks under $p = 0.9$. First columns correspond to **TI** and second column to **SB**. More results including performance in each domain and under different p can be found in Appendix B. Legend shown in first plot only.

5.1 MAIN RESULTS

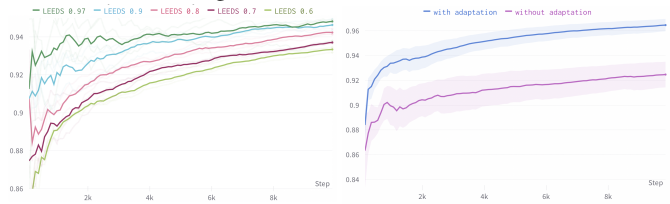
Results on Omniglot-MNIST-FashionMNIST (OMF). The online evaluations of the compared methods are shown in fig. 2 for non-stationary levels $p = 0.9$ and $p = 0.75$. For each setting we report separately the online accuracies on pre-training domain and on the other two OOD domains, to show how our method keeps improving on the OOD domains while also remembering the pre-tarining tasks. As shown in the plots, Our method LEEDS achieves superior performance compared to all other baseline algorithms in both settings. More specifically, on the IND domain all methods pre-trained using MAML perform similarly, but are outperformed by LEEDS and CMAML++ which can detect task boundaries. However, on the OOD domains our algorithm significantly outperforms all other baselines, including CMAML++. This is due to the key OOD adaptation module that allows LEEDS to dynamically adapt the meta model based on the task distribution. Interestingly, comparing the

performance for MAML and ANIL provides some insights on the limitations of re-using pre-trained representations in non-stationary environments. In fact, the ANIL baseline, which does not adapt its inner representations, performs poorly compared to MAML on the OOD domains, but achieves similar results on the pre-training domain. Also, the results highlight some limitations of the recently introduced FOML (Rajasegaran et al., 2022) method, which achieves lower performance than other competitive baselines. This is because FOML requires the tasks to be not mutually exclusive, which may not hold for the standard few-shot benchmarks considered in our experiments.

Results on Tiered-ImageNet (TI) and Synbols (SB) benchmarks. We report the online accuracies on all domains and on OOD domains for these two benchmarks in fig. 3. Due to space limitation, results for each domain are deferred to appendix B. Because the distribution of the pre-training tasks is similar to the OOD ones for the Tiered-ImageNet benchmark, methods such as MAML can perform reasonably well. In fact, in the lower non-stationary case ($p = 0.75$, can be found in Appendix B), MAML is able to outperform the more complex CMAML++ baseline. However, our algorithm still achieves the best performance under both non-stationary levels and in both benchmarks. Note that in the larger **TI** dataset case, the FOML algorithm, which stores all previously seen tasks, runs out of memory after around 6500 online episodes. Again because of similarity between OOD and IND tasks in the **TI** benchmark, static representations learned by ANIL are useful for all domains.

5.2 Ablation Studies

Task boundaries detection. The table in the right of fig. 4 provides the precision and recall scores of the task switch detection schemes for our method and CMAML++. Our detection scheme outperforms that of CMAML++ in all metrics. This is because, the detection scheme in CMAML++ is based on comparing successive losses, which could lead to over detection of task boundaries, especially when the task loss is too high at the first time the task is revealed to the online algorithm.



	Method	Prec.	Rec.
TI	LEEDS	**96.6**	**99.27**
	CMAML++	90.93	92.98
SB	LEEDS	**91.81**	**99.37**
	CMAML++	87.09	93.89

Figure 4: **Left:** LEEDS under different p. **Center:** LEEDS with and without domain adaptation.**Right:** Task boundaries detection on **Tiered-ImageNet (TI)** and **Synbols (SB).**

Importance of domain adaptation module. We investigate the importance of the distribution shift detection module that allows our algorithm LEEDS to update the meta model differently for in-distribution and out-of-distribution tasks. Fig. 4 shows the performance of our algorithm with and without the distribution shift detection module. The performance of the algorithm significantly improves ($\sim 4.3\%$ improvement) with this module. This shows that such a simple mechanism can effectively boost the online learning performance by allowing the agent to learn more from OOD data while also remembering pre-training knowledge.

Sensitivity to frequency of task switches. Fig. 4 shows the performance of our algorithm for different values of the probability p of task switches. The performance increases with p, which shows that our algorithm LEEDS can successfully re-use previous task knowledge to increase performance.

6 Conclusions

In this work, we study the online meta-learning problem in non-stationary environments without knowing the task boundaries. To address the problems therein, we propose LEEDS for efficient meta model and online task model updates. In particular, based two simple but effective detection mechanisms of the task switches and the distribution shift, LEEDS can efficiently reuse the best task model available without resetting to the meta model and distinguish the meta model updates for in-distribution tasks and out-of-distribution tasks so as to quickly learn the new knowledge for new distributions while preserving the old knowledge of the pre-training distribution. In particular, the meta model update in LEEDS is based on the current data only, eliminating the need of storing previous data. The theoretical analysis on the dynamic regret has clearly justified this design and extensive experiments corroborate the superior performance of LEEDS over related baseline methods on multiple benchmarks.

REFERENCES

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. 2009.

Durmus Alp Emre Acar, Ruizhao Zhu, and Venkatesh Saligrama. Memory efficient online meta learning. In *International Conference on Machine Learning*, pp. 32–42. PMLR, 2021.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Maria-Florina Balcan, Mikhail Khodak, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 424–433. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/balcan19a.html`.

Luca Bertinetto, Joao F Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR)*, 2018.

Massimo Caccia, Pau Rodríguez López, Oleksiy Ostapenko, Fabrice Normandin, Min Lin, Lucas Page-Caccia, Issam H. Laradji, Irina Rish, Alexandre Lacoste, David Vázquez, and Laurent Charlin. Online fast adaptation and knowledge accumulation (osaka): a new approach to continual learning. In *NeurIPS*, 2020.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Giulia Denevi, Dimitris Stamos, Carlo Ciliberto, and Massimiliano Pontil. Online-within-online meta-learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. *ArXiv*, abs/1710.11622, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. International Conference on Machine Learning (ICML)*, pp. 1126–1135, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pp. 1920–1930. PMLR, 2019.

James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3(2):97–139, 1957.

James Harrison, Apoorva Sharma, Chelsea Finn, and Marco Pavone. Continuous meta-learning without tasks. *Advances in neural information processing systems*, 33:17571–17581, 2020.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. *Advances in Neural Information Processing Systems*, 32, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, pp. 0. Lille, 2015.

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

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. *arXiv preprint arXiv:1905.05644*, 2019.

Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 7195–7201. IEEE, 2016.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pp. 2554–2563. PMLR, 2017.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv: Learning*, 2018.

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

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. *International Conference on Learning Representations (ICLR)*, 2019.

Jathushan Rajasegaran, Chesea Finn, and Sergey Levine. Fully online meta-learning without task boundaries. *arXiv preprint arXiv:2202.00263*, 2022.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 113–124, 2019.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2016.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850. PMLR, 2016.

Syed Shakib Sarwar, Aayush Ankit, and Kaushik Roy. Incremental learning in deep convolutional neural networks using partial network sharing. *IEEE Access*, 8:4615–4628, 2019.

Shai Shalev-Shwartz and Yoram Singer. Online learning: Theory, algorithms, and applications. 2007.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Huaxiu Yao, Yingbo Zhou, Mehrdad Mahdavi, Zhenhui Jessie Li, Richard Socher, and Caiming Xiong. Online structured meta-learning. *Advances in Neural Information Processing Systems*, 33: 6779–6790, 2020.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

Tianhe Yu, Xinyang Geng, Chelsea Finn, and Sergey Levine. Variable-shot adaptation for online meta-learning. *arXiv preprint arXiv:2012.07769*, 2020.

Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task agnostic continual learning using online variational bayes. *arXiv preprint arXiv:1803.10123*, 2018.

Pan Zhou, Xiaotong Yuan, Huan Xu, Shuicheng Yan, and Jiashi Feng. Efficient meta learning via minibatch proximal update. *Advances in Neural Information Processing Systems*, 32, 2019.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

Supplementary Material

We provide the details omitted in the main paper. The sections are organized as fellows:

• Appendix B: We provide more empirical results including final average accuracy of each of the methods and for all settings.

• Appendix C: We provide further details about datasets and baseline methods.

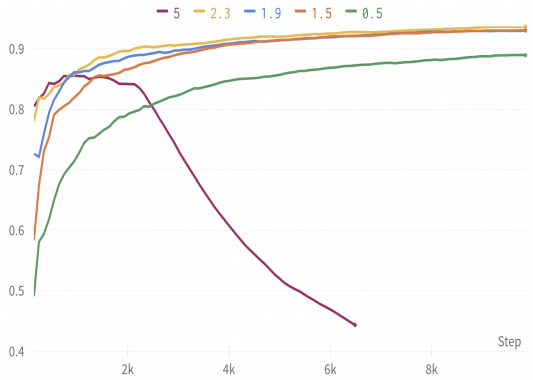
• Appendix D: We provide further experimental specifications and discuss the heuristics used to set the thresholds.

• Appendix F: We provide our proof of Theorem 1.

A SENSITIVITY TO THRESHOLDS ℓ AND τ AND TEMPERATURE δ

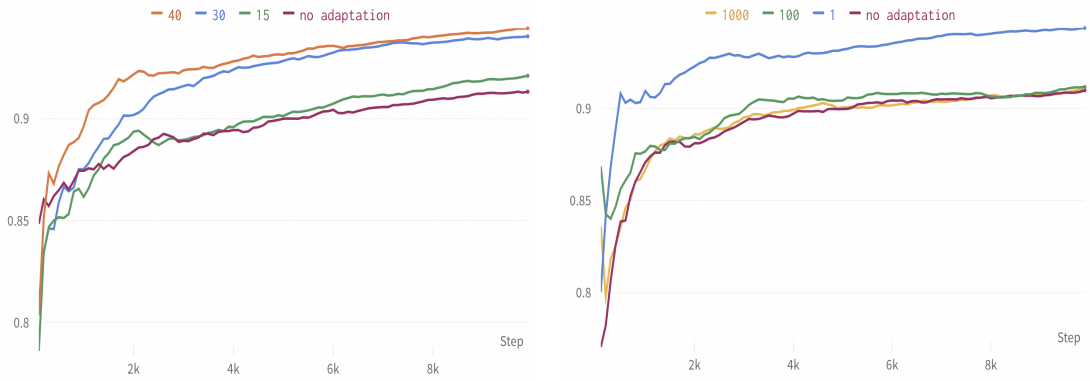
Figures 5 and 6 illustrate the sensitivity of our algorithm LEEDS with respect to the thresholds ℓ and τ and the temperature parameter δ in the energy-based detection module. As depicted in fig. 5, when the threshold ℓ is too small, the algorithm tends to over detect task switches (as indicated by low Recall for $\ell = 0.5$ in the table), which results in inferior performance of the algorithm due to ineffective reuse of task knowledge. On the other hand when ℓ is too large, the high misdetection rate (e.g., indicated by low Precision for $\ell = 5$) results in the algorithm mostly fine-tuning the online task model ϕ_t with the current task support data. As expected, this results in a failure mode (the algorithm diverges) due to the adversariality of different tasks. We find that values of ℓ in the rage [1.5, 2.3] yield the best performance of our algorithm.

Figure 6 (a) shows that larger values of τ, which collapse to updating the meta-model at each step (even for pretraining task distribution), does not substantially improve the performance. This demonstrates the advantage of the distinct meta-update scheme proposed for in- and out-of-distribution tasks, which avoids unnecessary frequent meta-updates for the pretraining tasks and thus allows a more judicious usage of computational budget. Lower values of τ (e.g. $\tau = 15$) tend to detect all task distributions as the pertaining one, and thus corresponds to eliminating the domain adaptation component of our algorithm. We also find that simply setting the temperature $\delta = 1$ in the energy expression yields the best performance, and large values of δ also eliminates the effectiveness of the energy-based detection module (fig. 6 (b)). This is in fact in accordance to the finding in Liu et al. (2020), which also suggests setting $\delta = 1$.



ℓ	Precision	Recall
$\ell = 0.5$	99.9	68.0
$\ell = 1.5$	99.9	98.4
$\ell = 1.9$	99.9	99.3
$\ell = 2.3$	99.9	99.7
$\ell = 5$	38.13	99.9

Figure 5: Performance of LEEDS for different values of the threshold ℓ. **Left plot:** Performance on all encountered domains during online learning. **Right table:** Task boundaries detection for different values of ℓ. Experiments are conducted on the Omniglot-MNIST-FashionMNIST benchmark.



(a) Sensitivity to energy threshold τ

(b) Sensitivity to temperature scale

Figure 6: Performance of LEEDS for: (a) different values of the energy threshold τ and (b) different scales of the temperature δ. For both plots we report the performance on all encountered domains during online learning. Experiments are conducted on the Omniglot-MNIST-FashionMNIST benchmark.

B More Experimental Results

In this section, we provide more experimental results for each of the benchmarks. Figures 7 and 8 show the online evaluations of the different methods for the **Tiered-ImageNet** and **Synbols** benchmarks under $\mathbf{p} = \mathbf{0.75}$. These additional results further show that our algorithm LEEDS outperforms other baseline algorithms in the OOD domains and at the same time also retains its performance on the pre-training tasks. The performance of the methods that do not adapt meta-parameters during online learning phase (such as MAML and ANIL) drops drastically when OOD tasks are far away from the pre-training ones (as shown in Table 1 for **Omniglot-MNIST-FashionMNIST**). For settings in which OOD tasks are close to the pre-training ones (such as in the **Tiered-ImageNet** dataset), MAML can perform similarly to CMAML++, as depicted in Table 2.

Further, Tables 1, 2, and 3 indicate that the performance of all compared baselines decrease as the non-stationarity level p decreases (smaller p indicates a higher task switch frequency). In fact, this is also captured by our theoretical upper bound for the dynamic regret in Theorem 1, as the accumulative model shift P_D is likely to increase when the environment becomes less stationary. Such intuition/result is also confirmed by the plots in Figure 9. As already shown in Figure 4 for all encountered domains, Figure 9 depicts the same dependency on p for the pre-training domain. The superior performance of LEEDS even for the pre-training domain particularly shows that the re-use of task knowledge is beneficial for online meta-learning, as opposed to the usual practice of "resetting" to meta-parameters at each step.

By comparing the two plots in Figure 10, it can be seen that the advantage of our domain adaptation module is more significant when the OOD domains are far away from the pre-training one, as is the case for the FashionMNIST OOD domain compared to the Omniglot pre-training domain.

C Further Descriptions about Datasets and Baseline Methods

C.1 Datasets

We study dynamic online meta-learning on the following benchmarks:

Omniglot-MNIST-FashionMNIST dataset. For this dataset, we consider 10-ways 5-shots classification tasks. We pre-train the meta model on a subset of the Omniglot dataset and then deploy it in the online learning environment where tasks are sampled from either the full Omniglot dataset, or from one of the OOD datasets, i.e., the MNIST or FashionMNIST datasets.

Tiered-ImageNet dataset. We consider 5-ways 5-shots classification tasks for this dataset. Following [cite paper], we split the original Tiered-ImageNet dataset into the pre-training domain and the OOD

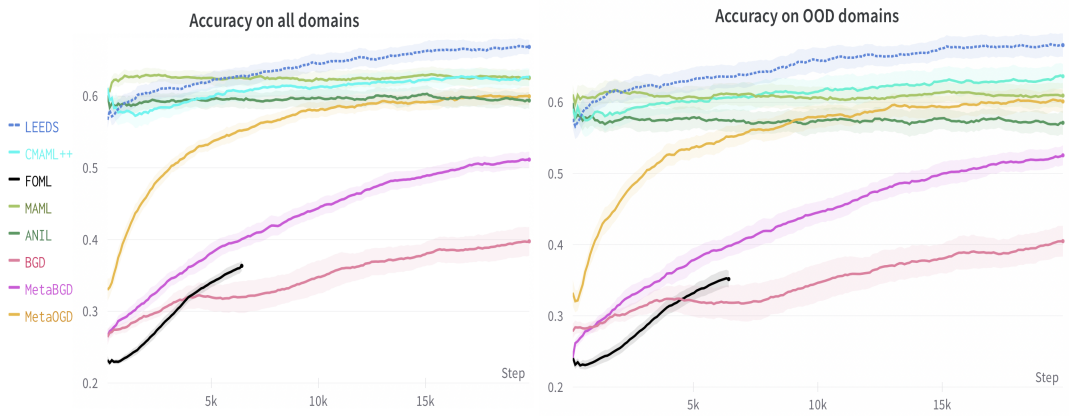


Figure 7: Online evaluations for the **Tiered-ImageNet (TI)** benchmark under $\mathbf{p = 0.75}$. **Left:** Accuracies on all encountered domains during online learning. **Right:** Accuracies on all encountered OOD domains during online learning. We compare all baselines on a 16GB GPU memory budget and FOML runs out of memory for this benchmark due to its linear growth in memory requirement.

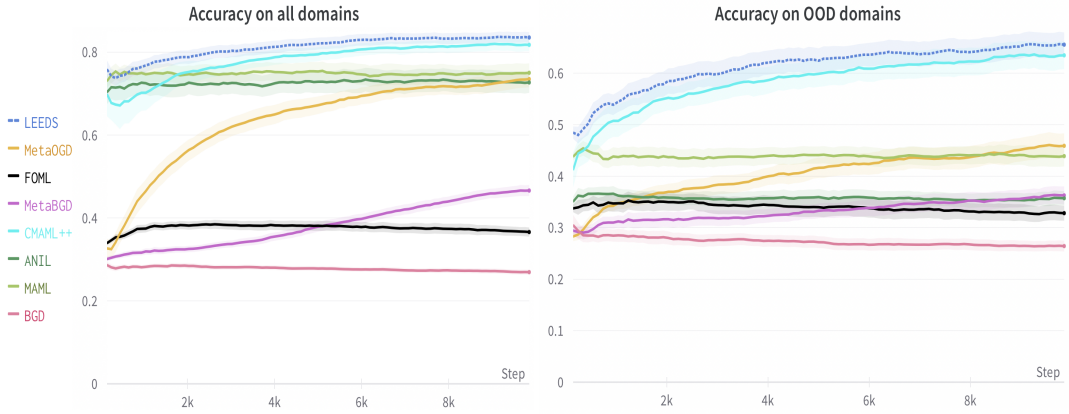


Figure 8: Online evaluations for the **Synbols (SB)** benchmark under $\mathbf{p = 0.75}$. **Left:** Accuracies on all encountered domains during online learning. **Right:** Accuracies on all encountered OOD domains during online learning.

domains. More specifically, we pre-train on the first 200 classes of the original training set, use the remaining classes as the first OOD domain (ood1), and set the original test classes as second OOD domain (ood2). During online learning phase, we set the full training set to be IND domain, and ood1 and ood2 as OOD domains. We evaluate algorithms over 20000 online learning episodes.

Synbols dataset. We consider 4-ways 4-shots classification tasks in this dataset. The meta model is pre-trained on characters from 3 different alphabets and deployed on characters from a new alphabet (ood1). We also consider font classification tasks as additional OOD tasks (ood2).

C.2 Baseline Methods

We compare our algorithm with the following baseline methods for online meta-learning.

(1) **C-MAML Caccia et al. (2020) and C-MAML++:** the continual MAML approach (C-MAML) pre-trains the meta model using MAML and employs an online learning strategy based on task boundaries detection. Since C-MAML does not evaluate the task models on separate query sets, for a fair comparison we adapt it to do so and call the resulting algorithm C-MAML++.

Method	non-stationarity $p = 0.9$			non-stationarity $p = 0.75$		
	pre-train	ood1	ood2	pre-train	ood1	ood2
LEEDS	**99.39** ± 0.09	**96.44** ± 0.11	**82.87** ± 0.19	**98.97** ± 0.10	**95.68** ± 0.12	**81.49** ± 0.22
CMAML++	98.78 ± 0.12	92.52 ± 0.19	76.16 ± 0.28	97.39 ± 0.11	89.07 ± 0.20	73.35 ± 0.35
CMAML	89.79 ± 0.54	84.06 ± 0.80	69.70 ± 0.63	75.51 ± 0.94	70.41 ± 1.22	58.58 ± 1.27
FOML	89.20 ± 0.61	70.84 ± 0.76	64.83 ± 0.74	81.68 ± 0.59	59.24 ± 0.78	58.07 ± 0.77
MAML	95.07 ± 0.10	62.02 ± 0.14	54.67 ± 0.17	95.51 ± 0.11	62.31 ± 0.13	54.83 ± 0.13
ANIL	96.54 ± 0.12	42.14 ± 0.16	40.12 ± 0.13	96.88 ± 0.11	42.08 ± 0.11	40.00 ± 0.13
MetaOGD	84.05 ± 1.66	73.73 ± 1.39	60.03 ± 1.60	85.67 ± 1.57	75.09 ± 1.43	60.51 ± 1.65
BGD	63.58 ± 2.25	46.12 ± 2.10	44.89 ± 1.41	23.86 ± 2.36	17.97 ± 2.17	19.81 ± 1.63
MetaBGD	77.73 ± 1.26	59.11 ± 0.84	54.67 ± 0.82	43.95 ± 1.31	27.87 ± 0.94	30.14 ± 0.98

Table 1: Average accuracy over 10000 online episodes on **Omniglot-MNIST-FashionMNIST** benchmark under different non-stationarity levels. "pre-train" domain: **Omniglot**; "ood1" domain: **MNIST**, "ood2" domain: **FashionMNIST**. The advantage of our algorithm LEEDS over the other baselines is more significant in the ood domains.

Method	non-stationarity $p = 0.9$		non-stationarity $p = 0.75$	
	all domains	ood domains	all domains	ood domains
LEEDS	**66.07** ± 0.24	**67.43** ± 0.38	**64.52** ± 0.17	**65.80** ± 0.31
CMAML++	63.83 ± 0.27	63.75 ± 0.55	61.28 ± 0.23	61.96 ± 0.41
FOML	35.90 ± 0.56	35.87 ± 0.83	32.02 ± 0.42	31.61 ± 0.69
MAML	62.37 ± 0.46	61.00 ± 0.72	62.54 ± 0.37	60.88 ± 0.65
ANIL	59.78 ± 0.21	57.61 ± 0.38	59.57 ± 0.22	57.38 ± 0.36
MetaOGD	57.01 ± 0.28	57.32 ± 0.66	56.80 ± 0.25	56.94 ± 0.42
BGD	40.95 ± 0.85	41.44 ± 1.15	35.48 ± 0.76	35.97 ± 1.09
MetaBGD	49.21 ± 1.05	50.01 ± 1.25	44.58 ± 1.12	45.30 ± 1.20

Table 2: Average accuracy over 20000 online episodes on **Tiered-ImageNet** benchmark under different non-stationarity levels. The different domains are distinct splits of the original Tiered-ImageNet dataset (please see experimental setup in Section 5 for details on how these splits are obtained).

Method	non-stationarity $p = 0.9$		non-stationarity $p = 0.75$	
	all domains	ood domains	all domains	ood domains
LEEDS	**85.12** ± 0.91	**67.48** ± 0.97	**82.22** ± 0.32	**63.68** ± 0.36
CMAML++	81.14 ± 1.05	62.39 ± 1.00	79.74 ± 1.07	60.70 ± 1.12
FOML	46.40 ± 0.61	41.73 ± 0.73	37.46 ± 0.27	34.13 ± 0.31
MAML	76.25 ± 0.63	42.70 ± 0.68	74.87 ± 0.42	43.84 ± 0.45
ANIL	64.58 ± 0.32	34.51 ± 0.54	72.69 ± 0.30	35.66 ± 0.49
MetaOGD	72.04 ± 0.67	46.69 ± 0.77	67.93 ± 0.59	42.66 ± 0.62
BGD	25.63 ± 0.07	25.61 ± 0.09	27.53 ± 0.08	27.17 ± 0.11
MetaBGD	53.74 ± 0.41	42.25 ± 0.52	40.79 ± 0.23	34.63 ± 0.33

Table 3: Average accuracy over 10000 online episodes on **Synbols** benchmark under different non-stationarity levels. The "pre-train" domain corresponds to 3 different alphabets from Synbols dataset, "ood1" corresponds to a new (w.r.t. the pre-training one) alphabet from Synbols dataset, and "ood2" contains font classification tasks.

(2) **FOML Rajasegaran et al. (2022):** the fully online meta-learning method updates online parameters using the latest online data and maintains a concurrent meta-training process to guide the online updates regularized by the meta model.

(3) **MAML Finn et al. (2017) and ANIL Raghu et al. (2019):** the MAML baseline consists of an offline pre-training phase and an online deployment phase. During offline pre-training the method

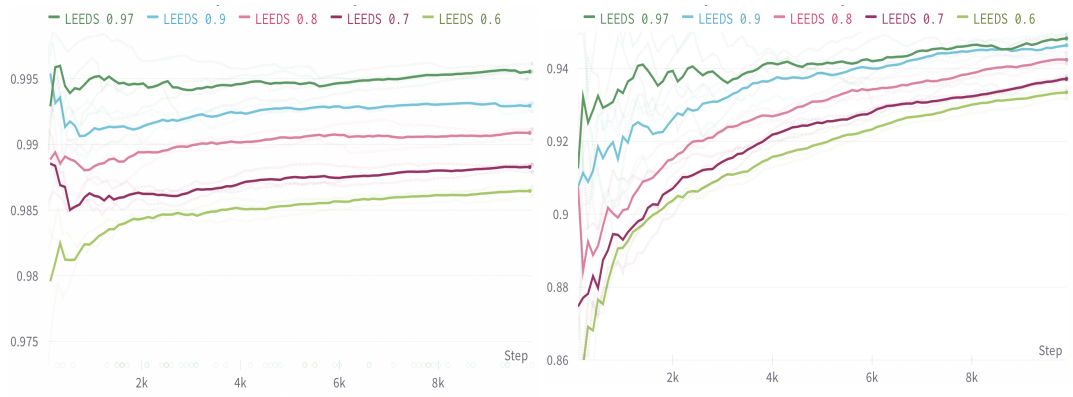


Figure 9: Performance of our algorithm LEEDS under different p. **Left plot:** Evaluations in pre-training domain. **Right plot:** Evaluations in all domains.

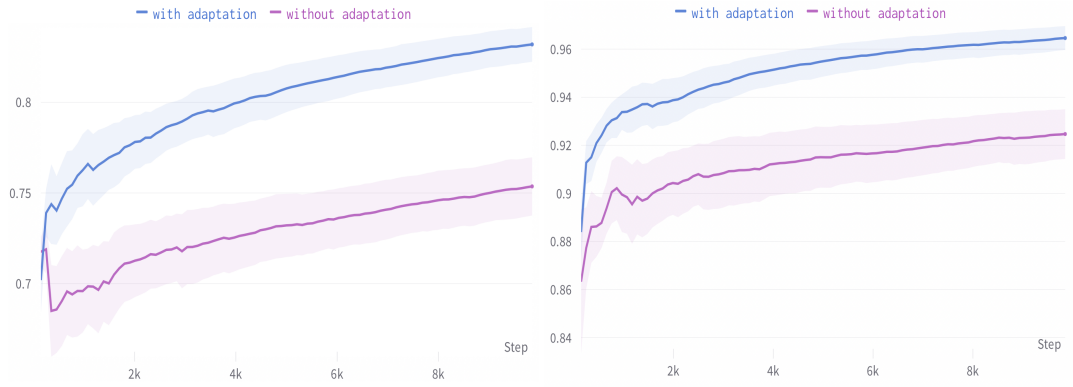


Figure 10: Performance of LEEDS with and without energy-based domain adaptation module. **Left plot:** Evaluations in OOD domain **FashionMINIST**. **Right plot:** Evaluations in OOD domain **MNIST**.

learns a common meta-initialization that will be used for all tasks, and the meta-initialization will never be updated at the online stage. The task model is adapted from the meta-initialization using the support set and evaluated on the query set. ANIL is similar to MAML but with partial parameter adaption, i.e., only the last layer is adapted for each task.

(4) **MetaBGD Caccia et al. (2020) and BGD Zeno et al. (2018):** the baseline MetaBGD combines MAML and the Bayesian gradient descent method during online learning.

(5) **MetaOGD Zinkevich (2003):** the meta online gradient descent method simply updates the meta-model at each step using a MAML-like meta-objective evaluated on the current task data.

D Further Experimental Details and Hyperameter Search

D.1 Further Experimental Specifications

In all our experiments, we consider classification tasks. The cross-entropy loss between predictions and true labels is used to train all models. We use the same convolutional neural network (CNN) architectures widely adopted in few-shot learning literature Finn et al. (2017; 2019); Caccia et al. (2020), which include four convolutional blocks followed by a linear classification layer. Each convolutional block is a stack of one 3×3 convolution layer followed by BatchNormalization, ReLU, and 2×2 MaxPooling layers. For the **Omniglot-MNIST-FashionMNIST** benchmark, we use 64 filters in each convolutional layer and downsample the gray-scale images to 28×28 spacial resolution

so as to have 64-dimensional feature vectors before classification. For **Tiered-ImageNet** and **Synbols** datasets, the inputs are respectively $3 \times 64 \times 64$ and $3 \times 32 \times 32$ RGB images, resulting respectively in 1024- and 256-dimensional feature vectors for 64 filters in each convolution layer.

D.2 Further Implementation Specifications

We will make all our codes publicly available after the review process. The implementation of FOML Rajasegaran et al. (2022) method is not released yet, and hence we compare with our own implementation of their algorithm. For all other baselines, we used their publicly available implementations. Codes for our algorithm LEEDS and all other baselines are provided in the supplementary materials of our submission. All codes are tested with Python 3.6 and Pytorch 1.2.

For example to run our algorithm LEEDS with the best hyperparameters that we obtained for the Omniglot-MNIST-FashionMNIST dataset under $p = 0.9$, one can run the following command:

```
python main.py --algo leeds --use_best 1
```

The experiment setting (e.g., the dataset to use) can be changed in configurations.yaml file. We run all methods on a single NVIDIA Tesla P100 GPU. All compared algorithms except FOML were able to run on 16GB GPU memory. FOML requires at least 32GB to reach 12000 online episodes for the Tiered-ImageNet dataset.

E Heuristics for Setting the Thresholds

For the energy threshold τ, we follow the strategy in Liu et al. (2020), i.e., we set the threshold τ using the pre-training tasks. More specifically, we set τ so that 95% of the pre-training inputs are correctly detected as pre-training data. In standard few-shot learning experimental setups, the true labeling for each individual task is usually randomly chosen. When there is a task switch, the task specific model learnt from the previous task generally does not fit the new task anymore, where the learning performance could be similar to that of a random model. Thus motivated, we find that a good heuristic to choose a starting value for the threshold ℓ is to use the loss value evaluated on a random model. For example, for 10-ways classification tasks that value would be $\ell_r = -\log(1/10) = 2.3$.

F Proof of Theorem 1

Consider the following algorithm for minimizing the regret in eq. (6), which updates the meta-parameters at each step based on current task only:

$$\theta_d^{k+1} := \theta_d^k - \alpha_d^k \nabla f_d^k \left(U_d^k(\theta_d^k)\right), \quad \text{with } \theta_{d+1}^1 := \theta_d^{K_d+1} \tag{8}$$

For convenience, we next re-state the result in Theorem 1.

Theorem 1 (Re-stated)**.** *Suppose that Assumptions 1 and 2 hold. Define the path length as* $P_D := \sum_{d=2}^{D} \|\theta_d^* - \theta_{d-1}^*\| + 1$ *and set the learning rate* $\alpha_d^k = \frac{1}{\sqrt{\sum_{d=1}^{D} K_d}}$*. Then the dynamic regret in eq.* (6) *is bounded as*

$$\mathrm{R}_{\mathrm{meta}}(\theta_1^*, ..., \theta_D^*) \leq \mathcal{O}\left(P_D \sqrt{\sum\nolimits_{d=1}^{D} K_d}\right). \tag{9}$$

Further, if $\alpha_d^k = \sqrt{\frac{P_D}{\sum_{d=1}^{D} K_d}}$*, then*

$$\mathrm{R}_{\mathrm{meta}}(\theta_1^*, ..., \theta_D^*) \leq \mathcal{O}\left(\sqrt{P_D \sum\nolimits_{d=1}^{D} K_d}\right). \tag{10}$$

Proof of Theorem 1. In the following, we let $h_d^k := f_d^k \circ U_d^k$ and define M to be such that $\left\|\theta_d^k\right\| \leq M$ for all d and k. Such an upper bound can be enforced by implementing the project gradient descent so that the model parameter is always updated within a compact set with a radius M.

First, note that for any given domain index d, the term $\sum_{k=1}^{K_d} f_d^k\left(\boldsymbol{U}_d^k\left(\theta_d^k\right)\right)-\sum_{k=1}^{K_d} f_d^k\left(\boldsymbol{U}_d^k(\theta_d^*)\right)$ in the dynamic regret in eq. (6) corresponds to a static regret defined over K_d steps for loss functions $h_d^k := f_d^k \circ U_d^k$. Hence, applying steps similar to those in the proof of Theorem 3.1 in Hazan et al. (2016) for static regret, we have

$$\left\|\theta_d^{k+1}-\theta_d^*\right\|^2 \leq \left\|\theta_d^k-\theta_d^*\right\|^2+\left(\alpha_d^k\left\|\nabla h_d^k(\theta_d^k)\right\|\right)^2-2\alpha_d^k \nabla h_d^k(\theta_d^k)^\top\left(\theta_d^k-\theta_d^*\right).$$

Using Assumption 1 and rearranging the terms, we obtain

$$2\nabla h_d^k(\theta_d^k)^\top\left(\theta_d^k-\theta_d^*\right) \leq \frac{\left\|\theta_d^k-\theta_d^*\right\|^2-\left\|\theta_d^{k+1}-\theta_d^*\right\|^2}{\alpha_d^k}+\alpha_d^k G^2.$$

Telescoping the above inequality from $k=1$ to K_d and using the convexity of h_d^k yield

$$\begin{aligned}\sum_{k=1}^{K_d} h_d^k(\theta_d^k)-\sum_{k=1}^{K_d} h_d^k(\theta_d^*) \leq & \frac{1}{2\alpha_d^1}\left\|\theta_d^1-\theta_d^*\right\|^2-\frac{1}{2\alpha_d^{K_d}}\left\|\theta_d^{K_d+1}-\theta_d^*\right\|^2 \\ & +\frac{1}{2}\sum_{k=2}^{K_d}\left(\frac{1}{\alpha_d^k}-\frac{1}{\alpha_d^{k-1}}\right)\left\|\theta_d^k-\theta_d^*\right\|^2+\frac{G^2}{2}\sum_{k=1}^{K_d}\alpha_d^k. \qquad (11)\end{aligned}$$

Hence, fixing the stepsizes to $\alpha_d^k=\frac{1}{\sqrt{\sum_{d=1}^D K_d}}$ we obtain

$$\begin{aligned}\sum_{k=1}^{K_d} h_d^k(\theta_d^k)-\sum_{k=1}^{K_d} h_d^k(\theta_d^*) \leq & \frac{\sqrt{\sum_{d=1}^D K_d}}{2}\left(\left\|\theta_d^1-\theta_d^*\right\|^2-\left\|\theta_d^{K_d+1}-\theta_d^*\right\|^2\right)+\frac{G^2 K_d}{2\sqrt{\sum_{d=1}^D K_d}} \\ = & \frac{\sqrt{\sum_{d=1}^D K_d}}{2}\left(\left\|\theta_d^1-\theta_d^*\right\|^2-\left\|\theta_{d+1}^1-\theta_d^*\right\|^2\right)+\frac{G^2 K_d}{2\sqrt{\sum_{d=1}^D K_d}}\end{aligned}$$

where the last equality applies the transition scheme $\theta_d^{K_d+1}=\theta_{d+1}^1$. Now, telescoping from $d=1$ to D, we obtain

$$\sum_{d=1}^D\left(\sum_{k=1}^{K_d} h_d^k(\theta_d^k)-\sum_{k=1}^{K_d} h_d^k(\theta_d^*)\right) \leq \frac{\sqrt{\sum_{d=1}^D K_d}}{2}\underbrace{\sum_{d=1}^D\left(\left\|\theta_d^1-\theta_d^*\right\|^2-\left\|\theta_{d+1}^1-\theta_d^*\right\|^2\right)}_{P}+\frac{G^2}{2}\sqrt{\sum_{d=1}^D K_d}. \qquad (12)$$

Next, we upper-bound the term P. We have

$$\begin{aligned}\left\|\theta_d^1-\theta_d^*\right\|^2-\left\|\theta_{d+1}^1-\theta_d^*\right\|^2 = & \left\|\theta_d^1\right\|^2-2\left\langle\theta_d^1, \theta_d^*\right\rangle-\left\|\theta_{d+1}^1\right\|^2+2\left\langle\theta_{d+1}^1, \theta_d^*\right\rangle \\ = & \left\|\theta_d^1\right\|^2-\left\|\theta_{d+1}^1\right\|^2+2\left\langle\theta_{d+1}^1, \theta_d^*\right\rangle-2\left\langle\theta_d^1, \theta_{d-1}^*\right\rangle+2\left\langle\theta_d^1, \theta_{d-1}^*\right\rangle \\ & -2\left\langle\theta_d^1, \theta_d^*\right\rangle\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}\sum_{d=1}^D\left(\left\|\theta_d^1-\theta_d^*\right\|^2-\left\|\theta_{d+1}^1-\theta_d^*\right\|^2\right) & \leq\left\|\theta_1^1\right\|^2-\left\|\theta_{D+1}^1\right\|^2+2\left\langle\theta_{D+1}^1, \theta_D^*\right\rangle+2\sum_{d=2}^D\left\langle\theta_d^1, \theta_d^*-\theta_{d-1}^*\right\rangle \\ & \leq M^2+2M^2+2M\sum_{d=2}^D\left\|\theta_d^*-\theta_{d-1}^*\right\| \\ & \leq 3M^2\left(1+\sum_{d=2}^D\left\|\theta_d^*-\theta_{d-1}^*\right\|\right)\end{aligned}$$

$$= 3M^2 P_D, \tag{13}$$

where the last equality follows from the definition of $P_D := \sum_{d=2}^{D} \left\|\theta_d^* - \theta_{d-1}^*\right\| + 1$. Hence, combining eq. (12) and eq. (13), we obtain

$$\begin{aligned} \mathrm{R}_{\mathrm{meta}}(\theta_1^*, ..., \theta_D^*) &\leq \frac{3M^2 P_D}{2} \sqrt{\sum_{d=1}^{D} K_d} + \frac{G^2}{2} \sqrt{\sum_{d=1}^{D} K_d} \\ &\leq \mathcal{O}\left(P_D \sqrt{\sum_{d=1}^{D} K_d} \right) \end{aligned} \tag{14}$$

which finishes the proof for the first result.

Similarly, setting $\alpha_d^k = \sqrt{\frac{P_D}{\sum_{d=1}^{D} K_d}}$, we obtain the following upper bound for the dynamic regret in eq. (6)

$$\mathrm{R}_{\mathrm{meta}}(\theta_1^*, ..., \theta_D^*) \leq \mathcal{O}\left(\sqrt{P_D \sum_{d=1}^{D} K_d} \right). \tag{15}$$

□