# MULTI-SUBSPACE STRUCTURED META-LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Meta-learning aims to extract meta-knowledge from historical tasks to accelerate learning on new tasks. A critical challenge in meta-learning is to handle task heterogeneity, i.e., tasks lie in different distributions. Unlike typical meta-learning algorithms that learn a globally shared initialization, recent structured meta-learning algorithms formulate tasks into multiple groups and learn an initialization for tasks in each group using centroid-based clustering. However, those algorithms still require task models in the same group to be close together and fail to take advantage of negative correlations between tasks. In this paper, task models are formulated into a subspace structure. We propose a MUlti-Subspace structured Meta-Learning (MUSML) algorithm to learn the subspace bases. We establish the convergence and analyze the generalization performance. Experimental results confirm the effectiveness of the proposed MUSML algorithm.

## 1 INTRODUCTION

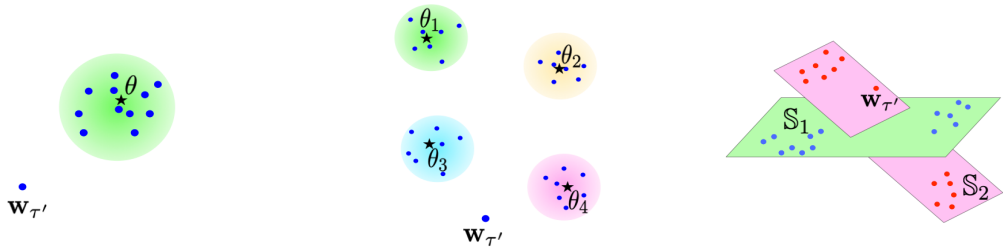
Humans are capable of learning new tasks from a few trials by taking advantage of prior experiences. However, the state-of-the-art performance of deep networks heavily relies on the availability of large amounts of labeled samples. To improve sample efficiency, *meta-learning* algorithms (Bengio et al., 1991; Thrun & Pratt, 1998) are designed to learn meta-knowledge from historical tasks and accelerate learning on unseen tasks. Meta-learning has been widely used for few-shot learning (Finn et al., 2017; Wang et al., 2020), neural architecture search (Zoph & Le, 2017; Liu et al., 2018), hyperparameter optimization (Maclaurin et al., 2015; Franceschi et al., 2018), reinforcement learning (Nagabandi et al., 2018; Rakelly et al., 2019), recommendation systems (Vartak et al., 2017; Lee et al., 2019a), and natural language processing (Gu et al., 2018; Obamuyide & Vlachos, 2019).

Typical meta-learning algorithms (Finn et al., 2017; Denevi et al., 2019; Rajeswaran et al., 2019; Zhou et al., 2019) learn a globally shared meta-model for all tasks. For example, the Model-Agnostic Meta-Learning (MAML) algorithm (Finn et al., 2017) learns a meta-initialization such that a good model for an unseen task can be fine-tuned from limited examples by a few gradient descent steps. However, when the tasks are heterogeneous, the task models are diverse and a common meta-model may not be sufficient.

To tackle this issue, recent works (Jerfel et al., 2019; Zhou et al., 2021) cluster tasks into multiple groups and learn an initialization for tasks in each group. Specifically, Jerfel et al. (2019) formulate the task distribution as a mixture of hierarchical Bayesian models and update the components (i.e., initializations) using Expectation Maximization. Zhou et al. (2021) first train task models using the vanilla MAML, and then cluster them into several groups based on the Euclidean distance. The cluster centroids become the group-specific initializations. However, centroid-based clustering fails to take advantage of negative correlation between tasks (e.g., $\mathbf{w}$ and $-\mathbf{w}$ may be assigned to different clusters) and fails to handle tasks that are distant from all clusters (e.g., tasks $\tau'$ in Figures 1(a) and 1(b)).

Another approach to deal with task heterogeneity is based on formulating task models into a subspace structure. Recent attempts (Kong et al., 2020; Tripuraneni et al., 2021) focus on the simple case where linear regression tasks are drawn from a single subspace. They leverage a moment-based estimator to recover its basis. However, it is not easy to extend such moment-based methods to nonlinear models (e.g., neural networks) or general losses (e.g., cross-entropy loss).

In this paper, we propose to learn multiple subspaces for nonlinear models or general losses, and treat the subspace bases as meta-parameters. For each task, the base learner selects a subspace that




(a) Single cluster. (b) Multiple clusters. (c) Subspace structure.

Figure 1: Different formulations of task structure.

the learned task model achieves the best performance on the training set. The meta-learner then updates the basis of the selected subspace by minimizing the validation loss of the learned task model. We establish convergence results for the proposed algorithm. We show theoretically that the expected generalization gap depends on complexity of the subspaces, while the expected excess risk depends on both complexity of the subspaces and distance between the optimal task models and the learned subspaces. Experiments on standard benchmark datasets verify the effectiveness of the proposed method.

In summary, the contributions of this paper are as follows:

(i) We propose a MUlti-Subspace structured Meta-Learning (MUSML) algorithm to learn multiple subspaces for task models. The proposed algorithm can be applied to both linear and nonlinear models.

(ii) We prove the convergence of MUSML and theoretically study the generalization performance.

(iii) Experimental results demonstrate that MUSML outperforms the state-of-the-arts.

**Notations:** Vectors (e.g., $\mathbf{x}$) and matrices (e.g., $\mathbf{X}$) are denoted by lowercase and uppercase boldface letters, respectively. For a vector $\mathbf{x}$, its $\ell_2$-norm is represented as $\|\mathbf{x}\|$. For a matrix $\mathbf{X}$, its $\ell_2$-norm is $\|\mathbf{X}\|$, and its Frobenius norm is $\|\mathbf{X}\|_{\mathrm{F}}$. Subspaces are denoted by blackboard boldface letters (e.g., $\mathbb{S}$). $\mathbf{1}_m \in \mathbb{R}^m$ denotes a $m$-dimensional vector with all entries being 1.

## 2 Related Work

Meta-learning designs algorithms to extract meta-knowledge from historical tasks so that new tasks can be learned fast with a few training examples. Popular meta-learning algorithms can be divided into three categories: metric-based approach (Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Bertinetto et al., 2018; Sung et al., 2018; Oreshkin et al., 2018; Lee et al., 2019b), memory-based approach (Santoro et al., 2016; Munkhdalai & Yu, 2017), and optimization-based approach (Ravi & Larochelle, 2017; Finn et al., 2017; Rajeswaran et al., 2019; Denevi et al., 2019; Balcan et al., 2019).

Most of meta-learning methods assume a globally shared meta-model (e.g., meta-initialization or meta-regularization) for all tasks. To tackle heterogeneous tasks, recent works (Vuorio et al., 2019; Yao et al., 2019a;b) tailor the meta-initialization to task representations, while Denevi et al. (2020) learn a meta-regularization conditioning on tasks' side information. Since good task representations or side information are not easy to obtain, Jerfel et al. (2019) and Zhou et al. (2021) propose to formulate tasks into multiple groups and parameters for tasks within the same group are assumed to be close in terms of the Euclidean distance. However, centroid-based methods still require parameters of related tasks to be close together and fail to handle negatively correlated tasks. To overcome these challenges, recent attempts (Kong et al., 2020; Saunshi et al., 2020; Tripuraneni et al., 2021) study linear regression tasks that are drawn from a low-dimensional subspace. Using a moment-based estimator, the basis can be recovered (Kong et al., 2020; Tripuraneni et al., 2021). However, their algorithms are limited to linear regression and a single subspace.

# 3 METHODOLOGY

## 3.1 PROBLEM FORMULATION

Let $p(\tau)$ be a task distribution. In meta-learning, a collection of tasks are used to learn meta-parameters. Each task $\tau \sim p(\tau)$ consists of a training set $\mathcal{D}_\tau^{tr} = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, N_{tr}\}$ and a validation set $\mathcal{D}_\tau^{vl} = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, N_{vl}\}$, where $\mathbf{x}$ are the features and $y$ the labels. Let $f(\cdot; \mathbf{w})$ be a model parameterized by $\mathbf{w} \in \mathbb{R}^d$, and $\mathcal{L}(\mathcal{D}; \mathbf{w}) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{D}} \ell(f(\mathbf{x}; \mathbf{w}), y)$ be the supervised loss on data set $\mathcal{D}$, where $\ell(\cdot, \cdot)$ is a loss function. The training set is used to learn the task model, while the validation set is used to update the meta-parameters at meta-training or evaluate the task model at meta-testing.

## 3.2 MULTI-SUBSPACE STRUCTURED META-LEARNING

In this paper, tasks are assumed to be clustered in multiple groups, and parameters for tasks in the same group lie in a low-dimensional subspace. Specifically, there are $K$ subspaces $\mathbb{S}_1, \ldots, \mathbb{S}_K$. For simplicity, we assume all subspaces to have the same dimensionality $m$. Let $\mathbf{S}_k \in \mathbb{R}^{d\times m}$ be a basis of $\mathbb{S}_k$. For a task $\tau$, the base learner selects the subspace $\mathbf{S}_{k_\tau}$ that $\tau$ lies in, and determines the linear combination weight $\mathbf{v}_\tau$ for the basis. The task model is then $\mathbf{w}_\tau = \mathbf{S}_{k_\tau}\mathbf{v}_\tau$. The basis set $\{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$ are meta-parameters to be updated by the meta-learner, while $(k_\tau, \mathbf{v}_\tau)$ are task parameters learned by the base learner.

Kong et al. (2020) and Tripuraneni et al. (2021) consider linear regression tasks and assume that the task parameters are drawn from a low-dimensional subspace. For this case, the column space of $\mathbb{E}_\tau \mathbb{E}_{(\mathbf{x},y)\sim\tau,(\mathbf{x}',y')\sim\tau} yy'\mathbf{x}\mathbf{x}'^\top$ is identical to the column space of $\mathbf{S}$. Using a moment-based estimator, they recover the basis from abundant meta-training tasks. However, their algorithms are infeasible for general models (e.g., neural networks) and general losses (e.g., cross-entropy loss).

For each task $\tau$, our base learner selects one of the $K$ subspaces such that the learned task model achieves the best performance on $\mathcal{D}_\tau^{tr}$. Specifically, the inner loop in meta-learning is formulated as

$$(k_\tau^*, \mathbf{v}_\tau^*) = \underset{k\in\{1,\ldots,K\},\mathbf{v}_\tau\in\mathbb{R}^m}{\arg\min} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_\tau), \tag{1}$$

where $\mathbf{S}_1, \ldots, \mathbf{S}_K$ are fixed. When $\ell(f(\mathbf{x}; \mathbf{w}), y)$ is convex in $\mathbf{w}$, $\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_\tau)$ is also convex in $\mathbf{v}_\tau$ and the minimization problem can be solved by convex programming (Boyd et al., 2004). However, for nonlinear models such as neural networks, the loss is usually non-convex and obtaining the global minimum of $\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_\tau)$ is intractable. Instead, for each $k$, we first compute $\mathbf{v}_{\tau,T_{\text{inner}}}^{(k)}$ by $T_{\text{inner}}$ gradient descent steps from an initial $\mathbf{v}_{\tau,0}^{(k)} = \frac{1}{m}\mathbf{1}_m$ with stepsize $\alpha$, then obtain the task parameters $k_\tau^* \equiv \arg\min_{1\le k\le K} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,T_{\text{inner}}}^{(k)})$ and $\mathbf{v}_\tau^* \equiv \mathbf{v}_{\tau,T_{\text{inner}}}^{(k_\tau^*)}$.

After obtaining the task parameters $(k_\tau^*, \mathbf{v}_\tau^*)$, the meta-learner updates $\mathbf{S}_{k_\tau^*}$ by a gradient descent step on the validation loss $\mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*)$. Let $\mathbf{w}_\tau^* \equiv \mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*$. As $\mathbf{v}_\tau^*$ is a function of $\mathbf{S}_{k_\tau^*}$, by the chain rule, $\nabla_{\mathbf{S}_{k_\tau^*}} \mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*) = \nabla_{\mathbf{w}_\tau^*}^\top \mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{w}_\tau^*) \nabla_{\mathbf{S}_{k_\tau^*}} \mathbf{w}_\tau^* = \nabla_{\mathbf{w}_\tau^*} \mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{w}_\tau^*)\mathbf{v}_\tau^{*\top} + \nabla_{\mathbf{w}_\tau^*}^\top \mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{w}_\tau^*)\mathbf{S}_{k_\tau^*} \nabla_{\mathbf{S}_{k_\tau^*}} \mathbf{v}_\tau^*$. Here, for simplicity, the dependence of $k_\tau^*$ on $\mathbf{S}_{k_\tau^*}$ is ignored. As $\mathbf{v}_\tau^* = \mathbf{v}_{\tau,0}^{(k_\tau^*)} - \alpha \sum_{t'=0}^{T_{\text{inner}}-1} \nabla_{\mathbf{v}_{\tau,t'-1}^{(k_\tau^*)}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau^*}\mathbf{v}_{\tau,t'-1}^{(k_\tau^*)})$, the total complexity of computing $\nabla_{\mathbf{S}_{k_\tau^*}} \mathbf{v}_\tau^*$ is $\mathcal{O}(T_{\text{inner}} dm^3)$. Usually, $m \ll d$ is very small.

The whole procedure, called MUSML, is shown in Algorithm 1. Similar to other optimization-based meta-learning algorithms (Finn et al., 2017; Rajeswaran et al., 2019), $T_{\text{inner}}$ is usually small (e.g., 1 to 5) for meta-training, but can be large (e.g., 10 to 20) at meta-testing.

## 3.3 THEORETICAL ANALYSIS

**Assumption 1.** (i) $\ell(f(\mathbf{x}; \mathbf{w}); y)$ is $\beta_1$-Lipschitz smooth[1] in $\mathbf{w}$; (ii) $\mathbf{v}_\tau^*$ is $\beta_2$-Lipschitz smooth in $\mathbf{S}_{k_\tau^*}$; (iii) $\mathbb{E}_\tau \|\nabla_{\mathbf{S}_{k_\tau^*}} \mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*) - \mathbb{E}_\tau \nabla_{\mathbf{S}_{k_\tau^*}} \mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*)\|^2 \le \sigma^2$; (iv) $\{\mathbf{v}_\tau^*, \tau \sim p(\tau)\}$

[1] In other words, $\|\nabla_{\mathbf{w}} \ell(f(\mathbf{x}; \mathbf{w}); y) - \nabla_{\mathbf{w}} \ell(f(\mathbf{x}; \mathbf{w}'); y)\| \le \beta_1 \|\mathbf{w} - \mathbf{w}'\|$.

**Algorithm 1** MUSML.

**Require:** stepsize $\alpha$, $\eta_t$, number of inner gradient steps $T_{\text{inner}}$, number of subspaces $K$, subspace dimension $m$;

**for** $t = 0, 1, \ldots, T-1$ **do**
  sample a task $\tau_t = (\mathcal{D}^{tr}_{\tau_t}, \mathcal{D}^{vl}_{\tau_t}) \sim p(\tau)$;
  base-learner:
  $\mathcal{L}^*_{\tau_t} = \infty$;
  **for** $k = 1, \ldots, K$ **do**
    $\mathbf{v}^{(k)}_{\tau_t,0} = \frac{1}{m}\mathbf{1}_m$;
    **for** $t' = 0, 1, \ldots, T_{\text{inner}} - 1$ **do**
      $\mathbf{v}^{(k)}_{\tau_t,t'+1} = \mathbf{v}^{(k)}_{\tau_t,t'} - \alpha \nabla_{\mathbf{v}^{(k)}_{\tau_t,t'}} \mathcal{L}(\mathcal{D}^{tr}_{\tau_t}; \mathbf{S}_{k,t}\mathbf{v}^{(k)}_{\tau_t,t'})$;
    **end for**
    **if** $\mathcal{L}(\mathcal{D}^{tr}_{\tau_t}; \mathbf{S}_{k,t}\mathbf{v}^{(k)}_{\tau_t,T_{\text{inner}}}) < \mathcal{L}^*_{\tau}$ **then**
      $k^*_{\tau_t} = k, \mathbf{v}^*_{\tau_t} = \mathbf{v}^{(k)}_{\tau_t,T_{\text{inner}}}, \mathcal{L}^*_{\tau_t} = \mathcal{L}(\mathcal{D}^{tr}_{\tau_t}; \mathbf{S}_{k,t}\mathbf{v}^{(k)}_{\tau_t,T_{\text{inner}}})$;
    **end if**
  **end for**
  meta-learner:
  $\mathbf{g}_t = \nabla_{\mathbf{S}_{k^*_{\tau_t},t}} \mathcal{L}(\mathcal{D}^{vl}_{\tau_t}; \mathbf{S}_{k^*_{\tau_t},t}\mathbf{v}^*_{\tau_t})$;
  **for** $k = 1, \ldots, K$ **do**
    $\mathbf{S}_{k,t+1} = \mathbf{S}_{k,t} - \eta_t I(k^*_{\tau_t}, k)\mathbf{g}_t$; ▷ $I(a,b) = 1$ if $a = b$ otherwise 0.
  **end for**
**end for**
**return** $\mathbf{S}_{1,T}, \ldots, \mathbf{S}_{K,T}$.

and basis vectors are in a compact set, and thus their $\ell_2$-norms are bounded by a constant $\beta_3 > 0$; (v) There exists a constant $\epsilon > 0$ such that for all $\tau \sim p(\tau)$, $\mathcal{L}(\mathcal{D}^{tr}_{\tau}; \mathbf{S}_{k^*_\tau}\mathbf{v}^*_\tau) \leq \min_{k \neq k^*_\tau, 1 \leq k \leq K} \mathcal{L}(\mathcal{D}^{tr}_{\tau}; \mathbf{S}_k \mathbf{v}^{(k)}_{\tau,T_{\text{inner}}}) - \epsilon$.

The assumptions on Lipschitz-smoothness and variance are commonly used in stochastic non-convex optimization (Ghadimi & Lan, 2013; Reddi et al., 2016) and meta-learning in non-convex settings (Fallah et al., 2020; Zhou et al., 2021), while the compactness assumption is also used in the convergence analysis of bilevel optimization (Franceschi et al., 2018).

Let $I(a, b) = 1$ if $a = b$, and 0 otherwise. The following Theorem establishes convergence of the proposed algorithm and the proof is similar to that in (Fallah et al., 2020). The $\mathcal{O}(1/\sqrt{T})$ rate is the same as MAML (Fallah et al., 2020; Ji et al., 2020) and other meta-learning algorithms (Zhou et al., 2019) . All proofs are in the Appendix.

**Theorem 1.** *Let* $\mathcal{L}_{meta}(\mathbf{S}_1, \ldots, \mathbf{S}_K) = \mathcal{L}(\mathcal{D}^{vl}_{\tau}; \mathbf{S}_{k^*_\tau}\mathbf{v}^*_\tau)$ *and* $\eta_t = \min\left(\frac{\epsilon}{2m\beta_1\beta_2\beta_3}, \frac{1}{\sqrt{T}}\right)$. *With Assumption 1, we have*

$$\min_{1 \leq t \leq T} \mathbb{E}\|\nabla_{[\mathbf{S}_{k,t}, \ldots, \mathbf{S}_{K,t}]} \mathcal{L}_{meta}(\mathbf{S}_{1,t}, \ldots, \mathbf{S}_{K,t})\|^2 \leq \mathcal{O}\left(\sum_{k=1}^{K} \frac{\sigma^2\sqrt{T}}{\mathbb{E}\sum_{t=1}^{T} I(k^*_{\tau_t}, k)}\right),$$

*where the expectation is over the random training samples. If each subspace is selected* $\frac{T}{K}$ *times in expectation, i.e.,* $\mathbb{E}\sum_{t=1}^{T} I(k^*_{\tau_t}, k) = \frac{T}{K}$ *for all* $k$*, then the upper bound simplifies to* $\mathcal{O}\left(\frac{\sigma^2 K^2}{\sqrt{T}}\right)$.

Next, we study the testing performance of the learned subspaces. The following assumption ensures that the task parameters are stable when one sample is changed in the training set. Stability is a widely-used tool to analyze the generalization of meta-learning algorithms and bilevel optimization (Maurer & Jaakkola, 2005; Bao et al., 2021).

**Assumption 2.** For any two training sets $\mathcal{D}^{tr}_1$ and $\mathcal{D}^{tr}_2$ that only differ in one sample, $\max_{1 \leq k \leq K} \|\arg\min_{\mathbf{v} \in \mathbb{R}^m} \mathcal{L}(\mathcal{D}^{tr}_1; \mathbf{S}_k\mathbf{v}) - \arg\min_{\mathbf{v} \in \mathbb{R}^m} \mathcal{L}(\mathcal{D}^{tr}_2; \mathbf{S}_k\mathbf{v})\| \leq \frac{K}{N_{tr}}$.

**Theorem 2.** *Let* $\tau'$ *be a testing task,* $\mathcal{R}(\tau'; \mathbf{S}_1, \ldots, \mathbf{S}_K) \equiv \mathbb{E}_{\mathcal{D}^{tr}_{\tau'} \sim \tau'} \mathbb{E}_{(\mathbf{x},y) \sim \tau'} \ell(f(\mathbf{x}; \mathbf{S}_{k^*_{\tau'}}\mathbf{v}^*_{\tau'}), y)$ *and* $\hat{\mathcal{R}}(\tau'; \mathbf{S}_1, \ldots, \mathbf{S}_K) \equiv \mathbb{E}_{\mathcal{D}^{tr}_{\tau'} \sim \tau'} \mathcal{L}(\mathcal{D}^{tr}_{\tau'}; \mathbf{S}_{k^*_{\tau'}}\mathbf{v}^*_{\tau'})$, *where* $(k^*_{\tau'}, \mathbf{v}^*_{\tau'}) =$

$\arg\min_{1\le k\le K,\mathbf{v}_{\tau'}\in\mathbb{R}^m} \mathcal{L}(\mathcal{D}^{tr}_{\tau'}; \mathbf{S}_k\mathbf{v}_{\tau'})$. *With Assumptions 1 and 2, (i) we have*

$$\mathcal{R}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) \le \hat{\mathcal{R}}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) + \mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}}\right); \tag{2}$$

*(ii) Let* $\mathbf{w}^o_{\tau'} \equiv \arg\min_{\mathbf{w}_{\tau'}\in\mathbb{R}^d} \mathbb{E}_{(\mathbf{x},y)\sim\tau'}\ell(f(\mathbf{x};\mathbf{w}_{\tau'}),y)$ *be the optimal model for* $\tau'$ *and* $\mathcal{R}^o(\tau') \equiv \mathbb{E}_{(\mathbf{x},y)\sim\tau'}\ell(f(\mathbf{x};\mathbf{w}^o_{\tau'}),y)$ *be its expected risk. Then,*

$$\mathcal{R}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) \le \mathcal{R}^o(\tau') + \mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}} + \min_{1\le k\le K} \mathit{dist}(\mathbf{w}^o_{\tau'},\mathbb{S}_k)\right), \tag{3}$$

*where* $\mathit{dist}(\mathbf{w}^o_{\tau'},\mathbb{S}_k) \equiv \min_{\mathbf{w}_{\tau'}\in\mathbb{S}_k} \|\mathbf{w}^o_{\tau'} - \mathbf{w}_{\tau'}\|$ *is the distance between* $\mathbf{w}^o_{\tau'}$ *and the subspace* $\mathbb{S}_k$.

Theorem 2 analyzes the effects of $m$ and $K$ to the testing performance. From (2), increasing the complexity of subspaces may reduce $\hat{\mathcal{R}}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K)$ but increase the expected generalization gap $\mathcal{R}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) - \hat{\mathcal{R}}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K)$. For fixed $K$ and $m$, proper subspaces enable the base learner to reduce both $\hat{\mathcal{R}}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K)$ and $\mathcal{R}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K)$. From (3), the expected excess risk $\mathcal{R}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) - \mathcal{R}^o(\tau')$ is upper bounded by $\mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}} + \min_{1\le k\le K} \mathrm{dist}(\mathbf{w}^o_{\tau'},\mathbb{S}_k)\right)$. The first term depends on the complexity of subspaces, while the second term arises from the approximation error of $\mathbf{w}^o_{\tau'}$ using the learned subspaces. Again, good subspaces reduce the excess risk.

For the centroid-based clustering method in (Zhou et al., 2021), its expected excess risk is bounded by $\mathcal{O}\left(\frac{\gamma^{T_{\text{inner}}}}{N_{tr}} + \|\boldsymbol{\omega}_{k^*_{\tau'}} - \mathbf{w}^o_{\tau'}\|^2\right)$, where $\gamma > 1$ and $\boldsymbol{\omega}_{k^*_{\tau'}}$ is the centroid of the cluster that $\tau'$ belongs to. The distance $\|\boldsymbol{\omega}_{k^*_{\tau'}} - \mathbf{w}^o_{\tau'}\|$ plays the same role as the term $\min_{1\le k\le K} \mathrm{dist}(\mathbf{w}^o_{\tau'},\mathbb{S}_k)$ in (3), which measures how far the optimal model $\mathbf{w}^o_{\tau'}$ is away from the subspaces or clusters.

### 3.4 A Practical Implementation for MUSML

The proposed MUSML is a model-agnostic meta-learning framework. Note that the basis incur additional memory cost, which can be problematic especially for deep networks that usually contain millions of parameters. As features extracted from the bottom layers are more general (Yosinski et al., 2014), a practical implementation is to divide the network weight $\mathbf{w}$ into two parts: (i) $\mathbf{w}^{(\text{btm})} \in \mathbb{R}^{d_{\text{btm}}}$ for the bottom layers near the input that are shared globally across all tasks, and (ii) $\mathbf{w}^{(\text{top})} \in \mathbb{R}^{d_{\text{top}}}$ for the top layers. Let the basis $\mathbf{S}_k$ be partitioned analogously as $[\mathbf{1}_m^\top \otimes \boldsymbol{\theta}; \mathbf{S}_k^{(\text{top})}]$, where $\boldsymbol{\theta} \in \mathbb{R}^{d_{\text{btm}}}$ is the globally shared parameters and $\otimes$ is the Kronecker product. This implementation reduces the memory from $\mathcal{O}(Kmd)$ to $\mathcal{O}(Kmd_{\text{top}} + d_{\text{btm}})$.

## 4 Experiments

### 4.1 Few-shot classification on Meta-Dataset

**Dataset**. We use the standard 5-way $N_{tr}$-shot setting ($N_{tr} = 1$ or 5) on the commonly-used *Meta-Dataset* benchmark (Yao et al., 2019a;b; Triantafillou et al., 2020) to evaluate the proposed method. This benchmark consists of 4 image classification datasets: *Caltech-UCSD Birds-200-2011* (denoted by *Bird*) (Welinder et al., 2010), *Describable Textures Dataset* (denoted by *Texture*) (Cimpoi et al., 2014) , *Fine-Grained Visual Classification of Aircraft* (denoted by *Aircraft*) (Maji et al., 2013), and *FGVCx-Fungi* (denoted by *Fungi*) (Schroeder & Cui, 2018). We adopt the split setting in (Yao et al., 2019a) that for each dataset, classes are randomly split into three parts for meta-training, meta-validation and meta-testing, respectively. Table 1 describes the statistics of this meta-dataset, and Figure 6 in the appendix shows some example images. Following (Yao et al., 2019a), each few-shot task samples classes from one of the four datasets.

**Network Architecture**. We use the Conv4 network in (Yao et al., 2019a;b), which has 4 modules. Each module is a $3 \times 3$ convolutional layer with 64 filters, followed by a batch normalization layer, ReLU activation, and a $2 \times 2$ max-pooling layer. Follow the practical implementation, the first two modules are shared globally by all the tasks. As the choice of the model $f(\mathbf{x};\mathbf{w})$ is flexible, a simple prototype classifier with the cosine similarity (Snell et al., 2017; Gidaris & Komodakis, 2018) is used here.

Table 1: Statistics for datasets.

| | dataset | #classes (meta-training/validation/testing) | #samples per class |
|---|---|---|---|
| *Meta-Dataset* | *Bird* (Welinder et al., 2010) | $64/16/20$ | 60 |
| | *Texture* (Cimpoi et al., 2014) | $30/7/10$ | 120 |
| | *Aircraft* (Maji et al., 2013) | $64/16/20$ | 100 |
| | *Fungi* (Schroeder & Cui, 2018) | $64/16/20$ | 150 |
| | *Mini-Imagenet* (Vinyals et al., 2016) | $64/16/20$ | 600 |

**Baselines**. We compare the proposed method with state-of-the-art baselines: (i) meta-learning algorithms with a globally shared meta-model including MAML (Finn et al., 2017), MetaSGD (Li et al., 2017), TapNet (Yoon et al., 2019), and ProtoNet (Snell et al., 2017); (ii) meta-learning algorithms with a task modulation network including TADAM (Oreshkin et al., 2018), MT-Net (Lee & Choi, 2018), BMAML (Yoon et al., 2018), and MMAML (Vuorio et al., 2019); (iii) structured meta-learning algorithms including HSML (Yao et al., 2019a), ARML (Yao et al., 2019b), TSA-MAML (Zhou et al., 2021), and (Jerfel et al., 2019) (denoted by DPMM).

**Implementation Details**. We use the cross-entropy loss for $\ell(\cdot, \cdot)$. For the base learner, we use the SGD optimizer with a learning rate of $0.1$. The number $T_{\text{inner}}$ of inner gradient steps is set to $3$ at the meta-training and $15$ at the meta-validation and meta-testing. We train the subspace bases for $30,000$ iterations using the Adam optimizer (Kingma & Ba, 2015) with an initial learning rate of $0.001$, which is then reduced by half every $5,000$ iterations. To prevent overfitting, we evaluate the performance on the meta-validation set every $1,000$ iterations and stop training when the meta-validation accuracy has no significant improvement for $10$ consecutive evaluations. By tuning the hyperparameters $K$ and $m$ from $\{1, 5, 10, 20, 30, 40\}$ using grid search, $(K = 5, m = 5)$ and $(K = 20, m = 5)$ achieve the highest meta-validation accuracy for the 1-shot and 5-shot settings, respectively, thus are used in experiments.

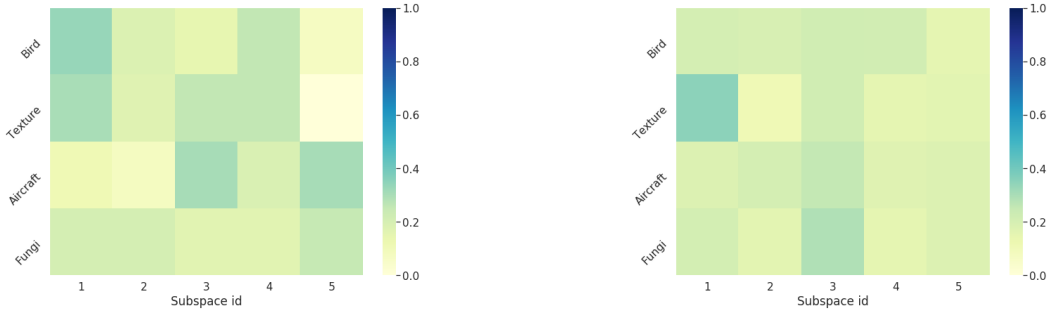
Table 2: Accuracies (with 95% confidence intervals) of 5-way 1-shot classification on *Meta-Dataset*. $^\dagger$ means that the result is obtained by running the code under this setting. Results of other baselines are from (Yao et al., 2019a;b).

| method | *Bird* | *Texture* | *Aircraft* | *Fungi* | average |
|---|---|---|---|---|---|
| MAML (Finn et al., 2017) | $53.94 \pm 1.45\%$ | $31.66 \pm 1.31\%$ | $51.37 \pm 1.38\%$ | $42.12 \pm 1.36\%$ | $44.77\%$ |
| MetaSGD (Li et al., 2017) | $55.58 \pm 1.43\%$ | $32.38 \pm 1.32\%$ | $52.99 \pm 1.36\%$ | $41.74 \pm 1.34\%$ | $45.67\%$ |
| ProtoNet$^\dagger$ (Snell et al., 2017) | $60.58 \pm 1.22\%$ | $34.48 \pm 1.18\%$ | $53.38 \pm 1.33\%$ | $40.61 \pm 1.27\%$ | $47.28\%$ |
| TapNet (Yoon et al., 2019) | $54.90 \pm 1.34\%$ | $32.44 \pm 1.23\%$ | $51.22 \pm 1.34\%$ | $42.88 \pm 1.35\%$ | $45.36\%$ |
| TADAM (Oreshkin et al., 2018) | $56.58 \pm 1.34\%$ | $33.34 \pm 1.27\%$ | $53.24 \pm 1.33\%$ | $43.06 \pm 1.33\%$ | $46.56\%$ |
| MT-Net (Lee & Choi, 2018) | $58.72 \pm 1.43\%$ | $32.80 \pm 1.35\%$ | $47.72 \pm 1.46\%$ | $43.11 \pm 1.42\%$ | $45.59\%$ |
| BMAML (Yoon et al., 2018) | $54.89 \pm 1.48\%$ | $32.53 \pm 1.33\%$ | $53.63 \pm 1.37\%$ | $42.50 \pm 1.33\%$ | $45.89\%$ |
| MMAML (Vuorio et al., 2019) | $56.82 \pm 1.49\%$ | $33.81 \pm 1.36\%$ | $53.14 \pm 1.39\%$ | $42.22 \pm 1.40\%$ | $46.50\%$ |
| DPMM$^\dagger$ (Jerfel et al., 2019) | $61.30 \pm 1.47\%$ | $35.21 \pm 1.35\%$ | $57.88 \pm 1.37\%$ | $43.81 \pm 1.45\%$ | $49.55\%$ |
| HSML (Yao et al., 2019a) | $60.98 \pm 1.50\%$ | $35.01 \pm 1.36\%$ | $57.38 \pm 1.40\%$ | $44.02 \pm 1.39\%$ | $49.35\%$ |
| ARML (Yao et al., 2019b) | $62.33 \pm 1.47\%$ | $35.65 \pm 1.40\%$ | $58.56 \pm 1.41\%$ | $44.82 \pm 1.38\%$ | $50.34\%$ |
| TSA-MAML$^\dagger$ (Zhou et al., 2021) | $61.37 \pm 1.42\%$ | $35.41 \pm 1.39\%$ | $58.78 \pm 1.37\%$ | $44.17 \pm 1.25\%$ | $49.93\%$ |
| MUSML (proposed) | $\mathbf{63.97 \pm 1.10\%}$ | $\mathbf{37.65 \pm 1.16\%}$ | $\mathbf{61.36\% \pm 1.20\%}$ | $\mathbf{46.23 \pm 1.12\%}$ | $\mathbf{52.30\%}$ |

**Results**. For each dataset, we report the classification accuracy averaged over $1,000$ tasks randomly sampled from the meta-testing set. The results are reported in Table 2 for the 1-shot setting and Table 3 for the 5-shot setting. As can be seen, in both settings, MUSML consistently outperforms current state-of-the-arts. Compared with ProtoNet, the better performance possessed by MUSML confirms the effectiveness of structuring task models intro multiple subspaces. Compared with other structured meta-learning methods (i.e., DPMM, HSML, ARML, and TSA-MAML), MUSML achieves higher accuracy.

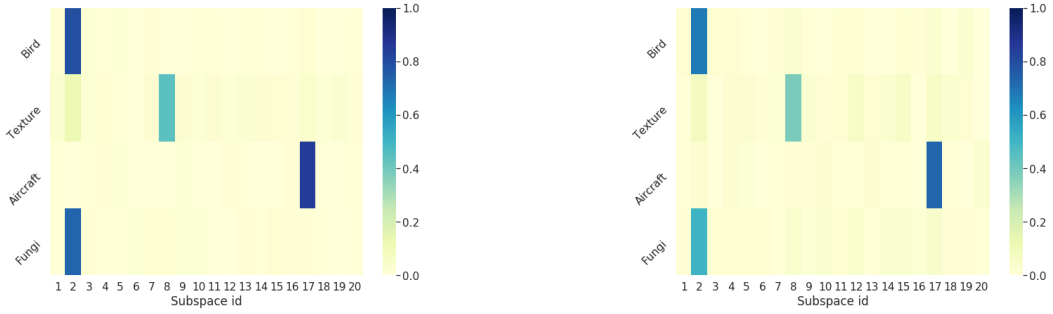
Table 3: Accuracies (with 95% confidence intervals) of 5-way 5-shot classification on *Meta-Dataset*. † means that the result is obtained by running the code under this setting. Results of other baselines are from (Yao et al., 2019a;b).

| method | *Bird* | *Texture* | *Aircraft* | *Fungi* | average |
|---|---|---|---|---|---|
| MAML (Finn et al., 2017) | $68.52 \pm 0.73\%$ | $44.56 \pm 0.68\%$ | $66.18 \pm 0.71\%$ | $51.85 \pm 0.85\%$ | 57.78% |
| MetaSGD (Li et al., 2017) | $67.87 \pm 0.74\%$ | $45.49 \pm 0.68\%$ | $66.84 \pm 0.70\%$ | $52.51 \pm 0.81\%$ | 58.18% |
| ProtoNet† (Snell et al., 2017) | $71.48 \pm 0.72\%$ | $50.36 \pm 0.67\%$ | $71.67 \pm 0.69\%$ | $55.68 \pm 0.82\%$ | 62.29% |
| TapNet (Yoon et al., 2019) | $69.07 \pm 0.74\%$ | $45.54 \pm 0.68\%$ | $67.16 \pm 0.66\%$ | $51.08 \pm 0.80\%$ | 58.21% |
| TADAM (Oreshkin et al., 2018) | $69.13 \pm 0.75\%$ | $45.78 \pm 0.65\%$ | $69.87 \pm 0.66\%$ | $53.15 \pm 0.82\%$ | 59.48% |
| MT-Net (Lee & Choi, 2018) | $69.22 \pm 0.75\%$ | $46.57 \pm 0.70\%$ | $63.03 \pm 0.69\%$ | $53.49 \pm 0.83\%$ | 58.08% |
| BMAML (Yoon et al., 2018) | $69.01 \pm 0.74\%$ | $46.06 \pm 0.69\%$ | $65.74 \pm 0.67\%$ | $52.43 \pm 0.84\%$ | 58.31% |
| MMAML (Vuorio et al., 2019) | $70.49 \pm 0.76\%$ | $45.89 \pm 0.69\%$ | $67.31 \pm 0.68\%$ | $53.96 \pm 0.82\%$ | 59.41% |
| DPMM† (Jerfel et al., 2019) | $72.22 \pm 0.70\%$ | $49.32 \pm 0.68\%$ | $73.55 \pm 0.69\%$ | $56.82 \pm 0.81\%$ | 63.00% |
| HSML (Yao et al., 2019a) | $71.68 \pm 0.73\%$ | $48.08 \pm 0.69\%$ | $73.49 \pm 0.68\%$ | $56.32 \pm 0.80\%$ | 62.39% |
| ARML (Yao et al., 2019b) | $73.68 \pm 0.70\%$ | $49.67 \pm 0.67\%$ | $74.88 \pm 0.64\%$ | $57.55 \pm 0.82\%$ | 63.95% |
| TSA-MAML† (Zhou et al., 2021) | $72.31 \pm 0.71\%$ | $49.50 \pm 0.68\%$ | $74.01 \pm 0.70\%$ | $56.95 \pm 0.80\%$ | 63.20% |
| MUSML (proposed) | $\mathbf{78.57 \pm 0.68\%}$ | $\mathbf{51.73 \pm 0.67\%}$ | $\mathbf{81.03 \pm 0.66\%}$ | $\mathbf{59.20 \pm 0.68\%}$ | **67.63%** |




(a) meta-training tasks. (b) meta-testing tasks.

Figure 2: Usage frequency of each learned subspace under the 5-way 1-shot setting ($K = 5, m = 5$). The value in the $(i, j)$-th grid is the frequency that MUSML assigns tasks from the $j$-th dataset to the $i$-th subspace. Dark colors indicate high frequencies.

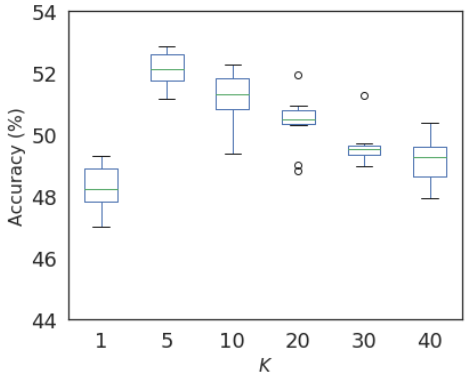



(a) meta-training tasks. (b) meta-testing tasks.

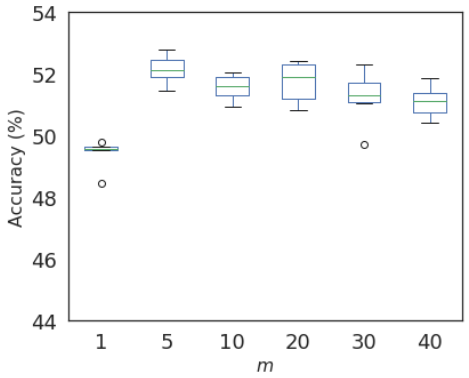
Figure 3: Usage frequency of each learned subspace for the 5-way 5-shot setting ($K = 20, m = 5$). The value in the $(i, j)$-th grid is the frequency that MUSML assigns tasks from the $j$-th dataset to the $i$-th subspace. Dark colors indicate high frequencies.

Figure 2 shows the usage frequency of learned subspaces under the 5-way 1-shot setting, and Figure 3 shows that under the 5-way 5-shot setting. As can be seen, the task structure under 5-shot setting is more clear. Figure 3 also reveals that 5-shot tasks from *Bird* and *Fungi* are prone to share the same subspace (i.e., the second subspace).

We further study the effects of $K$ and $m$ to the meta-testing accuracy. We repeat the experiment for 10 times and plot the accuracy in Figures 4 and 5 under the 1-shot and 5-shot settings, respectively.

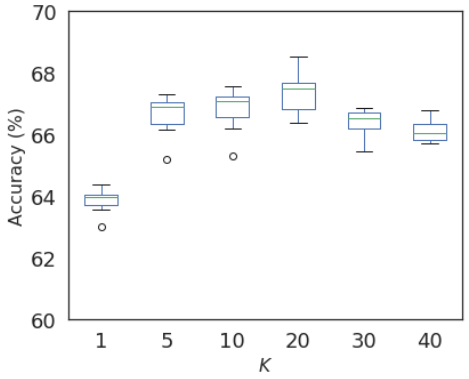



(a) Varying K when $m = 5$ is fixed.

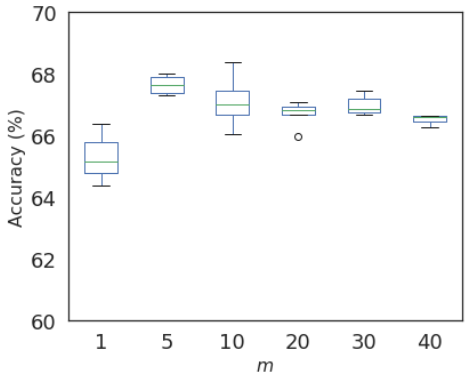



(b) Varying $m$ when $K = 5$ is fixed.

Figure 4: The meta-testing accuracy under the 5-way 1-shot setting.




(a) Varying K when $m = 5$ is fixed.




(b) Varying $m$ when $K = 20$ is fixed.

Figure 5: The meta-testing accuracy under the 5-way 5-shot setting.

As shown in Figures 4(a) and 5(a), a larger $K$ under the 1-shot setting is likely to cause overfitting than under the 5-shot setting. According to Figures 4(b) and 5(b), the accuracy is significantly improved when $m$ increases from 1 to 5.

We conduct experiments to verify that the effectiveness of MUSML is from its subspace structure instead of higher model complexity. We test the prototype classifier (Snell et al., 2017) with a $Kd \times$ wider network (denoted by Wide-ProtoNet). As shown in Table 4 and Table 5, MUSML still achieves better performance.

Table 4: Accuracies (with 95% confidence intervals) of 5-way 1-shot classification on *Meta-Dataset*.

| method | *Bird* | *Texture* | *Aircraft* | *Fungi* | average |
|---|---|---|---|---|---|
| ProtoNet (Snell et al., 2017) | $60.58 \pm 1.22\%$ | $34.48 \pm 1.18\%$ | $53.38 \pm 1.33\%$ | $40.61 \pm 1.27\%$ | 47.28% |
| Wide-ProtoNet | $61.58 \pm 1.10\%$ | $34.81 \pm 1.05\%$ | $57.41 \pm 1.28\%$ | $43.65 \pm 1.00\%$ | 49.36% |
| MUSML (proposed) | $\mathbf{63.97 \pm 1.10\%}$ | $\mathbf{37.65 \pm 1.16\%\%}$ | $\mathbf{61.36\% \pm 1.20\%}$ | $\mathbf{46.23 \pm 1.12\%}$ | **52.30%** |

Table 5: Accuracies (with 95% confidence intervals) of 5-way 5-shot classification on *Meta-Dataset*.

| method | *Bird* | *Texture* | *Aircraft* | *Fungi* | average |
|---|---|---|---|---|---|
| ProtoNet (Snell et al., 2017) | $71.48 \pm 0.72\%$ | $50.36 \pm 0.67\%$ | $71.67 \pm 0.69\%$ | $55.68 \pm 0.82\%$ | 62.29% |
| Wide-ProtoNet | $75.52 \pm 0.68\%$ | $50.49 \pm 0.58\%$ | $76.82 \pm 0.62\%$ | $57.12 \pm 0.71\%$ | 65.00% |
| MUSML (proposed) | $\mathbf{78.57 \pm 0.68\%}$ | $\mathbf{51.73 \pm 0.67\%}$ | $\mathbf{81.03 \pm 0.66\%}$ | $\mathbf{59.20 \pm 0.68\%}$ | **67.63%** |

### 4.2 FEW-SHOT CLASSIFICATION ON MINI-IMAGENET

**Experiment Setup**. We compare the proposed MUSML method with baselines on the *Mini-Imagenet* dataset (Vinyals et al., 2016), which consists of 100 randomly chosen classes from *ILSVRC*-2012 (Russakovsky et al., 2015). The statistics of this dataset are described in Table 1. We adopt the same split as (Ravi & Larochelle, 2017). All the methods in comparison use the experiment settings and the Conv4 backbone introduced in (Vinyals et al., 2016). The first two modules of Conv4 are shared globally by all the tasks. We evaluate the performance on the meta-validation set every $1,000$ iterations, and stop training when the meta-validation accuracy has no significant improvement for 10 consecutive evaluations. As this dataset has no explicitly heterogeneous structure, the complexity of subspace is probably low. We tune the hyperparameters $K$ and $m$ from $\{1, 2, 3, 4, 5\}$ using grid search, where $(k = 2, m = 3)$ and $(k = 3, m = 3)$ achieve the best meta-validation performance for the 1-shot and 5-shot settings, respectively, and thus use these settings.

**Result**. We report in Table 6 the classification accuracy averaged over 600 tasks randomly sampled from the meta-testing set. As can be seen, MUSML performs better than baselines.

Table 6: Accuracies (with 95% confidence intervals) of 5-way few-shot classification on the *Mini-Imagenet* dataset. "-" means that the corresponding result is not reported in related publications.

| method | 5-way 1-shot | 5-way 5-shot |
|---|---|---|
| MAML (Finn et al., 2017) | $48.7 \pm 1.8\%$ | $63.1 \pm 0.9\%$ |
| MetaSGD (Li et al., 2017) | $50.5 \pm 1.9\%$ | $64.0 \pm 0.9\%$ |
| ProtoNet (Snell et al., 2017) | $49.4 \pm 0.8\%$ | $68.2 \pm 0.7\%$ |
| TapNet (Yoon et al., 2019) | $50.7 \pm 0.1\%$ | $69.0 \pm 0.1\%$ |
| TADAM (Oreshkin et al., 2018) | $50.3 \pm 1.7\%$ | $66.2 \pm 0.8\%$ |
| MT-Net (Lee & Choi, 2018) | $51.7 \pm 1.8\%$ | - |
| BMAML (Yoon et al., 2018) | $50.0 \pm 1.9\%$ | - |
| MMAML (Vuorio et al., 2019) | $49.9 \pm 1.9\%$ | - |
| DPMM (Jerfel et al., 2019) | $49.3 \pm 1.5\%$ | $64.1 \pm 0.9\%$ |
| HSML (Yao et al., 2019a) | $50.4 \pm 1.8\%$ | - |
| ARML (Yao et al., 2019b) | $50.4 \pm 1.7\%$ | - |
| TSA-MAML (Zhou et al., 2021) | $49.5 \pm 1.3\%$ | $64.3 \pm 0.8\%$ |
| MUSML (proposed) | $\mathbf{54.1 \pm 1.0\%}$ | $\mathbf{69.9 \pm 0.7\%}$ |

## 5 CONCLUSION

In this paper, we proposed a novel algorithm called MUSML to learn multiple subspaces for task models. For each task, the base learner selects the subspace that the task lies in, and computes the corresponding linear combination weight. The subspace bases are meta-parameters updated by the meta-learner. We theoretically establish the convergence and analyze the generalization performance. Experimental results on benchmark datasets demonstrate that the proposed MUSML method outperforms the state-of-the-arts.

## ETHICS STATEMENT

We have read the ethics review guidelines and ensured that this paper conforms to them. No human subjects are researched in this work, so there is no such potential risk. All datasets used in the experiments are public and do not contain personally identifiable information or offensive content. There is no potential negative societal impacts.

## REPRODUCIBILITY STATEMENT

For all theoretical results, assumptions have been fully stated and complete proofs are provided in the Appendix. We have include code, data, and instructions needed to reproduce the main experimental results. All training details are mentioned in Section 4.

## REFERENCES


Maria-Florina Balcan, Mikhail Khodak, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning*, pp. 424–433. PMLR, 2019.

Fan Bao, Guoqiang Wu, Chongxuan Li, Jun Zhu, and Bo Zhang. Stability and generalization of bilevel programming in hyperparameter optimization. *arXiv preprint arXiv:2106.04188*, 2021.

Y. Bengio, S. Bengio, and J. Cloutier. Learning a synaptic learning rule. In *Proceedings of IJCNN-91-Seattle International Joint Conference on Neural Networks*, pp. 969 vol.2, 1991.

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

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *Proceedings of International Conference on Machine Learning*, pp. 1566–1575, 2019.

Giulia Denevi, Massimiliano Pontil, and Carlo Ciliberto. The advantage of conditional meta-learning for biased regularization and fine tuning. *Advances in Neural Information Processing Systems*, 33, 2020.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pp. 1082–1092. PMLR, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1126–1135, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pp. 1568–1577. PMLR, 2018.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3622–3631, 2018.

Ghassen Jerfel, Erin Grant, Tom Griffiths, and Katherine A Heller. Reconciling meta-learning and continual learning with online mixtures of tasks. In *Advances in Neural Information Processing Systems*, volume 32, pp. 9122–9133, 2019.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Multi-step model-agnostic meta-learning: Convergence and improved algorithms. *arXiv preprint arXiv:2002.07836*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, 2015.

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

Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. In *International Conference on Machine Learning*, pp. 5394–5404. PMLR, 2020.

Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1073–1082, 2019a.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019b.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2927–2936. PMLR, 2018.

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.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pp. 2113–2122. PMLR, 2015.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Andreas Maurer and Tommi Jaakkola. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6(6), 2005.

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

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2018.

Abiola Obamuyide and Andreas Vlachos. Model-agnostic meta-learning for relation classification with limited supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5873–5879, 2019.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 721–731, 2018.

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

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pp. 5331–5340. PMLR, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proceedings of International Conference on Learning Representations*, 2017.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pp. 314–323. PMLR, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

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.

Nikunj Saunshi, Yi Zhang, Mikhail Khodak, and Sanjeev Arora. A sample complexity separation between non-convex and convex meta-learning. In *International Conference on Machine Learning*, pp. 8512–8521. PMLR, 2020.

Brigit Schroeder and Yin Cui. FGVCx fungi classification challenge, 2018. URL `https://github.com/visipedia/fgvcx_fungi_comp`.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 4077–4087, 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.

Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pp. 3–17. Springer, 1998.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

Nilesh Tripuraneni, Chi Jin, and Michael Jordan. Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pp. 10434–10443. PMLR, 2021.

Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. *Advances in Neural Information Processing Systems*, 30:6904–6914, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 3630–3638, 2016.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. *Proceedings of Advances in Neural Information Processing Systems*, 32:1–12, 2019.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, pp. 7045–7054. PMLR, 2019a.

Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. Automated relational meta-learning. In *International Conference on Learning Representations*, 2019b.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7343–7353, 2018.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pp. 7115–7123. PMLR, 2019.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.

Pan Zhou, Xiaotong Yuan, Huan Xu, Shuicheng Yan, and Jiashi Feng. Efficient meta learning via minibatch proximal update. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 1534–1544, 2019.

Pan Zhou, Yingtian Zou, X Yuan, Jiashi Feng, Caiming Xiong, and SC Hoi. Task similarity aware meta learning: Theory-inspired improvement on maml. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2021.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

# A APPENDIX

## A.1 EXAMPLES FROM *Meta-Dataset*

(a) *Bird*. (b) *Texture*. (c) *Aircraft*. (d) *Fungi*.

Figure 6: Some examples taken from *Meta-Dataset*.

## A.2 PROOF OF THEOREM 1

*Proof.* Let $\mathrm{vec}(\mathbf{X})$ be the vectorization of a matrix $\mathbf{X}$, i.e., the column vector obtained by stacking the columns of $\mathbf{X}$. We first consider the $k$th subspace $\mathbb{S}_k$. As both $\ell(f(\mathbf{x};\mathbf{w}),y)$ and $\mathbf{v}_k^*$ are Lipschitz-smooth, $\mathcal{L}(\mathcal{D}_\tau^{vl};\mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*)$ is also Lipschitz smooth in $\mathbf{S}_{k_\tau^*}$. As $\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*) \leq \min_{k\neq k_\tau^*,1\leq k\leq K}\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,T_{\text{inner}}}^{(k)}) - \epsilon$, by the mean value theorem, $\mathcal{L}(\mathcal{D}_{\tau_t}^{(tr)};\mathbf{S}_{k_{\tau_t}^*,t+1}\mathbf{v}_{\tau_t}^*) = \mathcal{L}(\mathcal{D}_{\tau_t}^{(tr)};\mathbf{S}_{k_{\tau_t}^*,t}\mathbf{v}_{\tau_t}^*) + \eta_t \,\mathrm{vec}\left(\nabla_{\mathbf{S}_{k_{\tau_t}^*}}\mathcal{L}(\mathcal{D}_{\tau_t}^{(tr)};\mathbf{S}_{k_{\tau_t}^*}\mathbf{v}_{\tau_t}^*)\,|_{\mathbf{S}_{k_{\tau_t}^*}=\mathbf{S}_{k_{\tau_t}^*,\boldsymbol{\xi}}}\right)^\top \mathrm{vec}(\mathbf{S}_{k_\tau^*,t+1} - \mathbf{S}_{k_{\tau_t}^*,t}) \leq \mathcal{L}(\mathcal{D}_{\tau_t}^{(tr)};\mathbf{S}_{k_{\tau_t}^*,t}\mathbf{v}_{\tau_t}^*) + 2m\eta_t\beta_1\beta_2\beta_3 \leq \min_{k\neq k_{\tau_t}^*,1\leq k\leq K}\mathcal{L}(\mathcal{D}_{\tau_t}^{tr};\mathbf{S}_{k,t}\mathbf{v}_{\tau_t,T_{\text{inner}}}^{(k)})$, where $\mathbf{S}_{k_{\tau_t}^*,\boldsymbol{\xi}} \in [\mathbf{S}_{k_{\tau_t}^*,t+1},\mathbf{S}_{k_{\tau_t}^*,t}]$, thus, $k_{\tau_t}^*$ is unchanged within one meta step. Let $k_t \equiv k_{\tau_t}^*$ for notation simplicity. Using the Taylor expansion, it follows that

$$
\begin{aligned}
&\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t+1})\\
&\leq \mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}) + \mathrm{vec}(\nabla_{\mathbf{S}_{k,t}}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}))^\top \mathrm{vec}\left(\mathbf{S}_{k,t+1} - \mathbf{S}_{k,t}\right) + \frac{\beta_1\beta_2}{2}\|\mathbf{S}_{k,t+1} - \mathbf{S}_{k,t}\|^2\\
&= \mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}) - I(k_t,k)\eta_t \,\mathrm{vec}(\nabla_{\mathbf{S}_{k,t}}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}))^\top \mathrm{vec}(\nabla_{\mathbf{S}_{k,t}}\mathcal{L}(\mathcal{D}_{\tau_t}^{vl};\mathbf{S}_{k,t}\mathbf{v}_{\tau_t}^*))\\
&\quad + I(k_t,k)\frac{\eta_t^2\beta_1\beta_2}{2}\|\nabla_{\mathbf{S}_{k,t}}\mathcal{L}(\mathcal{D}_{\tau_t}^{vl};\mathbf{S}_{k,t}\mathbf{v}_{\tau_t}^*)\|^2\\
&\leq \mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}) - I(k_t,k)\eta_t \,\mathrm{vec}(\nabla_{\mathbf{S}_{k,t}}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}))^\top \mathrm{vec}\left(\nabla_{\mathbf{S}_{k,t}}\mathcal{L}(\mathcal{D}_{\tau_t}^{vl};\mathbf{S}_{k,t}\mathbf{v}_{\tau_t}^*) - \nabla_{\mathbf{S}_{k,t}}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t})\right)\\
&\quad + I(k_t,k)\eta_t\|\nabla_{\mathbf{S}_{k,t}}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t})\|^2 + I(k_t,k)\frac{\eta_t^2\beta_1\beta_2}{2}\|\nabla_{\mathbf{S}_{k,t}}\mathcal{L}(\mathcal{D}_{\tau_t}^{vl};\mathbf{S}_{k,t}\mathbf{v}_{\tau_t}^*)\|^2.
\end{aligned}
$$

Take conditional expectation w.r.t. $\mathbf{S}_{k,t}$ on both sides, then take the total expectation, we have

$$
\begin{aligned}
\mathbb{E}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t+1}) &\leq \mathbb{E}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}) - \mathbb{E}I(k_t,k)\eta_t(1-\frac{\eta_t\beta_1\beta_2}{2})\|\nabla_{\mathbf{S}_k}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t})\|^2 + \mathbb{E}I(k_t,k)\frac{\beta_1\beta_2\sigma^2\eta_t^2}{2}\\
&\leq \mathbb{E}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t}) - \mathbb{E}I(k_t,k)\frac{\eta_t}{2}\|\nabla_{\mathbf{S}_k}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t})\|^2 + \mathbb{E}I(k_t,k)\frac{\eta_t^2\beta_1\beta_2\sigma^2}{2}, \qquad (4)
\end{aligned}
$$

where we have used $1-\frac{\eta_t\beta_1\beta_2}{2} \geq \frac{1}{2}$ to obtain (4). Summing the above inequality over $t$, and rearranging it, we obtain

$$
\left(\frac{\eta_t}{2}\mathbb{E}\sum_{t=1}^{T}I(k_t,k)\right)\min_{1\leq t\leq T}\mathbb{E}\|\nabla_{\mathbf{S}_k}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t})\|^2 \leq \mathbb{E}\mathcal{L}_{\text{meta}}(\mathbf{S}_{k,0}) + \frac{\eta_t^2\beta_1\beta_2\sigma^2}{2}\mathbb{E}\sum_{t=1}^{T}I(k_t,k). \qquad (5)
$$

Dividing both sides by $\frac{\eta_t}{2}\mathbb{E}\sum_{t=1}^{T} I(k_t,k)$, as $\eta_t = \min\left(\frac{\epsilon}{2m\beta_1\beta_2\beta_3}, \frac{1}{\sqrt{T}}\right)$, we obtain

$$\min_{1\le t\le T} \mathbb{E}\|\nabla_{\mathbf{S}_{k,t}} \mathcal{L}_{\text{meta}}(\mathbf{S}_{k,t})\|^2 \le \mathcal{O}\left(\frac{\sigma^2\sqrt{T}}{\mathbb{E}\sum_{t=1}^{T} I(k_t,k)}\right), \tag{6}$$

and conclude that

$$\min_{1\le t\le T} \mathbb{E}\|\nabla_{[\mathbf{S}_{k,t},\ldots,\mathbf{S}_{K,t}]} \mathcal{L}_{\text{meta}}(\mathbf{S}_{1,t},\ldots,\mathbf{S}_{K,t})\|^2 \le \mathcal{O}\left(\sum_{k=1}^{K}\frac{\sigma^2\sqrt{T}}{\mathbb{E}\sum_{t=1}^{T} I(k^*_{\tau_t},k)}\right). \tag{7}$$

If $\mathbb{E}\sum_{t=1}^{T} I(k_t,k) = \frac{T}{K}$, then

$$\min_{1\le t\le T} \mathbb{E}\|\nabla_{[\mathbf{S}_{1,t},\ldots,\mathbf{S}_{K,t}]} \mathcal{L}_{\text{meta}}(\mathbf{S}_{1,t},\ldots,\mathbf{S}_{K,t})\|^2 = \mathcal{O}\left(\frac{\sigma^2 K^2}{\sqrt{T}}\right). \tag{8}$$

□

## B PROOF OF THEOREM 2

*Proof.* For notation simplicity, we omit the superscript of $\tau'$, and let $\mathbf{z} = (\mathbf{x}, y)$ denote samples. (i) Let $\mathbf{v}^*_{\tau,k} = \arg\min_{\mathbf{v}_\tau} \mathcal{L}(\mathcal{D}^{tr}_\tau; \mathbf{S}_k\mathbf{v}_\tau)$. We aim to show that the expected generalization gap $\mathcal{R}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) - \hat{\mathcal{R}}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) = \mathbb{E}_\tau\mathbb{E}_{\mathcal{D}^{tr}_\tau}\left[\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x};\mathbf{S}_k\mathbf{v}^*_{\tau,k}),y) - \mathcal{L}(\mathcal{D}^{tr}_\tau;\mathbf{S}_k\mathbf{v}^*_{\tau,k})\right]$ is bounded by an order of $\mathcal{O}(\frac{\lambda_K\sqrt{m}}{N_{tr}})$. Let $\mathcal{D}^{tr(i)}_\tau$ be a training set only differs with $\mathcal{D}^{tr}_\tau$ in the $i$th sample, ie, $\mathcal{D}^{tr(i)}_\tau \equiv (\mathcal{D}^{tr}_\tau - \{\mathbf{z}_i\}) \cup \{\mathbf{z}'_i\}$. And let $\mathbf{v}^{*(i)}_{\tau,k} \equiv \arg\min_{\mathbf{v}_\tau} \mathcal{L}(\mathcal{D}^{tr(i)}_\tau;\mathbf{S}_k\mathbf{v}_\tau)$. We will show that the solution obtained by minimizing the losses on $\mathcal{D}^{tr}_\tau$ is close to the one on $\mathcal{D}^{tr(i)}_\tau$: 1) $\mathbb{E}_{\mathcal{D}^{tr}_\tau}\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x};\mathbf{S}_k\mathbf{v}^*_{\tau,k}),y) = \frac{1}{N_{tr}}\sum_{i=1}^{N_{tr}}\mathbb{E}_{\mathcal{D}^{tr}_\tau}\mathbb{E}_{\mathbf{z}'_i\sim\tau}\ell(f(\mathbf{x}'_i;\mathbf{S}_k\mathbf{v}^*_{\tau,k}),y'_i) = \frac{1}{N_{tr}}\sum_{i=1}^{N_{tr}}\mathbb{E}_{\mathcal{D}^{tr(i)}_\tau}\mathbb{E}_{\mathbf{z}_i\sim\tau}\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}^{*(i)}_{\tau,k}),y_i) = \frac{1}{N_{tr}}\sum_{i=1}^{N_{tr}}\mathbb{E}_{\mathcal{D}^{tr}_\tau}\mathbb{E}_{\mathbf{z}'_i\sim\tau}\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}^{*(i)}_{\tau,k}),y_i)$; 2) $\mathbb{E}_{\mathcal{D}^{tr}_\tau}\mathcal{L}(\mathcal{D}^{tr}_\tau;\mathbf{S}_k\mathbf{v}^*_{\tau,k}) = \frac{1}{N_{tr}}\sum_{i=1}^{N_{tr}}\mathbb{E}_{\mathcal{D}^{tr}_\tau}\mathbb{E}_{\mathbf{z}'_i\sim\tau}\mathcal{L}(\mathcal{D}^{tr}_\tau;\mathbf{S}_k\mathbf{v}^*_{\tau,k})$. Hence, the expected generalization gap satisfies

$$\begin{aligned}
&\left|\mathbb{E}_{\mathcal{D}^{tr}_\tau}\left[\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x};\mathbf{S}_k\mathbf{v}^*_{\tau,k}),y) - \mathcal{L}(\mathcal{D}^{tr}_\tau;\mathbf{S}_k\mathbf{v}^*_{\tau,k})\right]\right| \\
&\le \frac{1}{N_{tr}}\sum_{i=1}^{N_{tr}}\mathbb{E}_{\mathcal{D}^{tr}_\tau}\mathbb{E}_{\mathbf{z}'_i\sim\tau}\left|\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}^{*(i)}_{\tau,k}),y_i) - \ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}^*_{\tau,k}),y_i)\right| \\
&\le \frac{\beta_1}{N_{tr}}\sum_{i=1}^{N_{tr}}\|\mathbf{S}_k\mathbf{v}^{*(i)}_{\tau,k} - \mathbf{S}_k\mathbf{v}^*_{\tau,k}\| && \text{(Lipschitz)} \\
&\le \frac{\beta_1}{N_{tr}}\sum_{i=1}^{N_{tr}}\|\mathbf{S}_k\|_{\text{F}}\|\mathbf{v}^{*(i)}_{\tau,k} - \mathbf{v}^*_{\tau,k}\| \\
&\le \frac{\beta_1\beta_3\sqrt{m}}{N_{tr}}\sum_{i=1}^{N_{tr}}\|\mathbf{v}^{*(i)}_{\tau,k} - \mathbf{v}^*_{\tau,k}\| \\
&\le \frac{\beta_1\beta_3 K\sqrt{m}}{N_{tr}}, && \text{(Assumption 2)}
\end{aligned}$$

where $\beta_1$ is the Lipschitz constant of $\ell(f(\mathbf{x};\mathbf{w}),y)$, $\beta_4$ is the bound of basis vectors as they stay in a compact set, and $\|\mathbf{X}\|_{\text{F}}$ is the Frobenius norm. As the above analysis is independent of the choice of $k$, we conclude that

$$\mathcal{R}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) - \hat{\mathcal{R}}(\tau';\mathbf{S}_1,\ldots,\mathbf{S}_K) \le \mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}}\right). \tag{9}$$

(ii) Let $\mathbf{w}_\tau^o = \arg\min_{\mathbf{w}_\tau} \mathbb{E}_{\mathbf{z}\sim\tau}\ell(\mathbf{z};\mathbf{w}_\tau)$ and $k_\tau^o = \arg\min_{1\le k\le K} \text{dist}(\mathbf{w}_\tau^o, \mathbb{S}_k)$. The excess risk satisfies

$$
\begin{aligned}
0 &\le \mathcal{R}(\tau';\mathbf{S}_1,\dots,\mathbf{S}_K) - \mathcal{R}^o(\tau') \\
&\le \underbrace{\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x};\mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*),y) - \mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*)\right]}_{\le\mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}}\right)} \\
&\quad + \mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\frac{1}{N_{tr}}\sum_{\mathbf{z}\in\mathcal{D}_\tau^{tr}} \ell(f(\mathbf{x};\mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*),y) - \mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x};\mathbf{w}_\tau^o),y)\right] \\
&\le \mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}}\right) + \underbrace{\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^*) - \mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{w}^o_{\tau,\mathbf{S}_{k_\tau^o}})\right]}_{\mathbf{S}_{k_\tau^*}\mathbf{v}_\tau^* \text{ is the optimal solution in the subspaces, thus, this term}\le 0} \\
&\quad + \mathbb{E}_{\mathcal{D}_\tau^{tr}}\|\nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_\tau^{tr};\boldsymbol{\xi}_\tau)\|\|\mathbf{w}^o_{\tau,\mathbf{S}_{k_\tau^o}^\perp}\| \qquad \text{(by the mean aalue theorem)} \\
&\le \mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}} + \beta_2\|\mathbf{w}_\tau^o\|_{\mathbf{S}_{k_\tau^o}^\perp}\right)
\end{aligned}
$$

where we have decomposed $\mathbf{w}_\tau^o = \mathbf{w}^o_{\tau,\mathbf{S}_{k_\tau^o}} + \mathbf{w}^o_{\tau,\mathbf{S}_{k_\tau^o}^\perp}$, $\boldsymbol{\xi}_\tau \in [\mathbf{w}^o_{\tau,\mathbf{S}_{k_\tau^o}}, \mathbf{w}_\tau^o]$, and the last inequality follows by the Lipschitz-smoothness. We conclude that

$$
\mathcal{R}(\tau';\mathbf{S}_1,\dots,\mathbf{S}_K) - \mathcal{R}^o(\tau') \le \mathcal{O}\left(\frac{K\sqrt{m}}{N_{tr}} + \min_{1\le k\le K}\text{dist}(\mathbf{w}^o_{\tau'},\mathbb{S}_k)\right).
$$

□