Subspace Learning for Effective Meta-Learning

Weisen Jiang [1 2] **James T. Kwok** [2] **Yu Zhang** [1 3]

Abstract

Meta-learning aims to extract meta-knowledge from historical tasks to accelerate learning on new tasks. Typical meta-learning algorithms like MAML learn a globally-shared meta-model for all tasks. However, when the task environments are complex, task model parameters are diverse and a common meta-model is insufficient to capture all the meta-knowledge. To address this challenge, in this paper, task model parameters are structured into multiple subspaces, and each subspace represents one type of meta-knowledge. We propose an algorithm to learn the meta-parameters (i.e., subspace bases). We theoretically study the generalization properties of the learned subspaces. Experiments on regression and classification meta-learning datasets verify the effectiveness of the proposed algorithm.

1. Introduction

Humans are capable of learning new tasks from 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 data efficiency, *meta-learning* (or *learning-to-learn*) (Bengio et al., 1991; Thrun & Pratt, 1998) seeks to design algorithms that extract meta-knowledge from historical tasks to accelerate learning on unseen tasks. Meta-learning has been widely used for few-shot learning (Finn et al., 2017; Wang et al., 2020b), 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), 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 samples by a few gradient updates. However, when the task environments are heterogeneous, task model parameters are diverse and a single meta-model may not be sufficient to capture all the meta-knowledge.

To tackle this issue, a variety of methods have been proposed to learn structured meta-knowledge by exploring the task structure (Jerfel et al., 2019; Yao et al., 2019; 2020; Wang et al., 2020a; Zhou et al., 2021a; Kong et al., 2020; Tripuraneni et al., 2021). For example, Jerfel et al. (2019) formulate the task distribution as a mixture of hierarchical Bayesian models, and update the components (i.e., initializations) using an Expectation Maximization procedure. TSA-MAML (Zhou et al., 2021a) first trains task models using vanilla MAML. Tasks are grouped into clusters by k-means clustering, and cluster centroids form group-specific initializations.

Alternatively, task model parameters can be formulated into a subspace. In the linear regression setting where task model vectors are sampled from a low-dimensional subspace, recent attempts (Kong et al., 2020; Tripuraneni et al., 2021) use a moment-based estimator to recover the subspace based on the property that the column space of the sample covariance matrix recovers the underlying subspace. However, for nonlinear models such as deep networks, this nice property no longer holds and the moment-based methods cannot be generalized.

In this paper, we propose a model-agnostic algorithm called MUSML (MUltiple Subspaces for Meta-Learning). Each subspace represents one type of meta-knowledge, and subspace bases are treated as meta-parameters. For each task, the base learner builds a task model from each subspace. The meta-learner then updates the subspace bases by minimizing a weighted validation loss of the task models. We theoretically establish upper bounds on the population risk, empirical risk and generalization gap. All these bounds depend on the complexity of the subspace mixture (number of component subspaces and subspace dimensionality). Ex-

[1]Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology [2]Department of Computer Science and Engineering, Hong Kong University of Science and Technology [3]Peng Cheng Laboratory. Correspondence to: Yu Zhang <yu.zhang.ust@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

periments on various datasets verify the effectiveness of the proposed MUSML.

Our major contributions are four-fold: (i) We formulate task model parameters into a subspace mixture and propose a novel algorithm to learn the subspace bases. (ii) The proposed MUSML is model-agnostic, and can be used on linear and nonlinear models. (iii) We theoretically study the generalization properties of the learned subspaces. (iv) We perform extensive experiments on synthetic and real-world datasets. Results on the synthetic dataset confirm that MUSML is able to discover the underlying subspaces of task model parameters. Results on the real–world datasets demonstrate superiority of MUSML over 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 ℓ_2-norm is $\|\mathbf{x}\|$. For a matrix $\mathbf{X}$, its spectral norm is $\|\mathbf{X}\|$ and its Frobenius norm is $\|\mathbf{X}\|_{\mathrm{F}}$. Subspaces (e.g., $\mathbb{S}$) are denoted by blackboard boldface letters. $\mathcal{U}(a, b)$ is the uniform distribution over the interval $[a, b]$. $\mathcal{N}(\mu; \sigma^2)$ is the univariate normal distribution with mean μ and variance σ^2, while $\mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\Sigma})$ is a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $\sigma_{\min}(\mathbf{X})$ is the smallest singular value of matrix $\mathbf{X}$.

2. Related Work

Meta-Learning. Popular meta-learning algorithms can be roughly divided into three categories: metric-based (Vinyals et al., 2016; Snell et al., 2017; Bertinetto et al., 2018; Oreshkin et al., 2018; Lee et al., 2019), memory-based (Santoro et al., 2016; Munkhdalai & Yu, 2017), and optimization-based. In this paper, we focus on the last category. Representative algorithms like MAML (Finn et al., 2017) and its variants (such as REPTILE (Nichol et al., 2018), ANIL (Raghu et al., 2020), Proto-MAML (Triantafillou et al., 2020), and BMG (Flennerhag et al., 2022)) first learn a meta-initialization from historical tasks. For a new task with limited examples, a good model is then obtained by a few gradient updates. Alternatively, other optimization-based algorithms such as iMAML (Rajeswaran et al., 2019), (Denevi et al., 2018; 2019), Meta-MinibatchProx (Zhou et al., 2019), and MetaProx (Jiang et al., 2021)) learn a meta-regularizer to bias risk minimization in the base learners.

Optimization-based meta-learning algorithms are effective when the task models are close together. However, real-world environments are complex, and the task model parameters are usually diverse. A globally-shared meta-model may not be sufficient for all tasks to achieve fast adaption.

To tackle this challenge, Vuorio et al. (2019) and Yao et al. (2019; 2020) build task-specific initializations by incorporating task representation. Denevi et al. (2020) propose to learn a meta-regularization conditioned on the task's side information. Since discriminative task representations and additional side information may not be easy to obtain, Jerfel et al. (2019) and Zhou et al. (2021a) cluster tasks into multiple groups and learn group-specific initializations. Recently, Kong et al. (2020); Saunshi et al. (2020); Tripuraneni et al. (2021) structure task model parameters to a subspace. However, this is limited to the linear regression setting and cannot be extended to nonlinear models such as deep networks.

Learning neural network subspaces. Recent studies (Li et al., 2018; Izmailov et al., 2020; Gressmann et al., 2020; Wortsman et al., 2021) show that deep network can be optimized in a low-dimensional parameter subspace. Wortsman et al. (2021) empirically demonstrates that a learned subspace contains diverse solutions that can be ensembled to boost accuracy. In meta-learning, learning of subspaces is more challenging since the optimization problem is bilevel and the learned subspaces need to generalize well on unseen tasks with limited samples.

3. Methodology

Let $p(\tau)$ be a task distribution. Each task $\tau \sim p(\tau)$ corresponds to a data distribution over $(\mathbf{x}, y)$, with input $\mathbf{x}$ and label y. In practice, this data distribution is only accessible via a training set $\mathcal{D}_\tau^{tr} = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, N_{tr}\}$ and a validation set $\mathcal{D}_\tau^{vl} = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, N_{vl}\}$ of i.i.d. samples. 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 its loss on $\mathcal{D}$, where $\ell(\cdot, \cdot)$ is a loss function (e.g., cross entropy or mean squared error). In meta-learning, a collection of tasks sampled from $p(\tau)$ are used to learn the meta-parameter. The base learner takes a task τ and meta-parameter to construct the model parameter $\mathbf{w}_\tau$. The meta-learner then minimizes the validation loss $\mathbb{E}_{\tau\sim p(\tau)}\mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{w}_\tau)$ w.r.t. the meta-parameter. In this paper, the task model parameters $\mathbf{w}_\tau$'s are assumed to form a subspace mixture, and the component subspace bases are treated as meta-parameters.

3.1. Linear Regression Tasks

We first focus on the linear setting, where the model is linear and all task parameters lie in one single (linear) subspace. The following Proposition shows that the underlying subspace can be recovered using a moment-based estimator.

Proposition 3.1. *(Kong et al., 2020; Tripuraneni et al., 2021). Assume that $p(\tau)$ is a distribution of linear regression tasks. Each task τ is associated with a $\mathbf{w}_\tau^\star \in \mathbb{R}^d$, and its samples are generated as $y = \mathbf{x}^\top \mathbf{w}_\tau^\star + \xi$, where $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\xi \sim \mathcal{N}(0, \sigma_\xi^2)$ is the noise. Then, $\mathbb{E}_{\tau\sim p(\tau)}\mathbb{E}_{(\mathbf{x},y)\sim\tau,(\mathbf{x}',y')\sim\tau} yy'\mathbf{x}\mathbf{x}'^\top = \mathbb{E}_{\tau\sim p(\tau)}\mathbf{w}_\tau^\star\mathbf{w}_\tau^{\star\top}$.*

Proposition 3.1 shows that $\hat{\mathbf{S}} \equiv \frac{1}{|\mathcal{B}|}\sum_{\tau\in\mathcal{B}} y_\tau y'_\tau \mathbf{x}_\tau \mathbf{x}'^\top_\tau$ is

an unbiased estimator of $\mathbb{E}_{\tau\sim p(\tau)}\mathbf{w}_\tau^\star\mathbf{w}_\tau^{\star\top}$, where $(\mathbf{x}_\tau, y_\tau)$ and $(\mathbf{x}'_\tau, y'_\tau)$ are two samples drawn from τ, and $\mathcal{B}$ is a collection of tasks. Hence, the column space of $\hat{\mathbf{S}}$ recovers the column space of $\mathbb{E}_{\tau\sim p(\tau)}\mathbf{w}_\tau^\star\mathbf{w}_\tau^{\star\top}$ (i.e., the underlying subspace) when the number of tasks is sufficient.

3.2. Proposed Method

While Proposition 3.1 can be used to recover the column space in a linear meta-learning setting, extension to the nonlinear setting (such as deep networks) is difficult. To address this problem, we propose a model-agnostic algorithm called MUSML (MUltiple Subspaces for Meta-Learning). We assume that the model parameters $\mathbf{w}_\tau$'s lie in K subspaces $\{\mathbb{S}_1, \ldots, \mathbb{S}_K\}$, which can be seen as an approximation to a nonlinear manifold. For simplicity, we assume that all K subspaces have the same dimensionality m (this can be easily extended to the case where the subspaces have different dimensionalities). Let $\mathbf{S}_k \in \mathbb{R}^{d\times m}$ be a basis of $\mathbb{S}_k$. $\{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$ are then the meta-parameters to be learned.

The proposed procedure is shown in Algorithm 1. Given a task τ, the base learner searches for the model parameter $\mathbf{w}_\tau$ over all subspaces with fixed $\mathbf{S}_k$ (steps 4-11). In each subspace $\mathbb{S}_k$, we search for the best linear combination $\mathbf{v}_{\tau,k}^\star$ of the subspace's basis to form $\mathbf{w}_\tau$ as

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

$\mathbf{S}_k\mathbf{v}_{\tau,k}^\star$ is then the task model parameter corresponding to the kth subspace. When $\ell(f(\mathbf{x};\mathbf{w}), y)$ is convex in $\mathbf{w}$, it is easy to verify that $\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_\tau)$ is also convex in $\mathbf{v}_\tau$. Hence, problem (1) can be solved as a convex program (Boyd & Vandenberghe, 2004). However, for nonlinear models such as deep networks, the loss function in (1) is nonconvex, and thus finding $\mathbf{v}_{\tau,k}^\star$ is computationally intractable. Instead, we seek an approximate minimizer $\mathbf{v}_{\tau,k}$ by performing T_{in} gradient descent steps from an initialization $\mathbf{v}_{\tau,k}^{(0)}$, i.e., $\mathbf{v}_{\tau,k}^{(t'+1)} = \mathbf{v}_{\tau,k}^{(t')} - \alpha\nabla_{\mathbf{v}_{\tau,k}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')})$ (where $\alpha > 0$ is the step size), and $\mathbf{v}_{\tau,k} \equiv \mathbf{v}_{\tau,k}^{(T_{in})}$.

At meta-training, one can assign τ to the subspace with the best training set performance. However, this is inefficient for learning meta-parameters since only one subspace is updated at each step. Similar to DARTS (Liu et al., 2018), we relax the categorical choice to a softmax selection over all candidate subspaces. The relaxed operation is differentiable and all subspace bases can then be updated simultaneously, which accelerates learning. Let $o_{\tau,k} = \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k})$ be the training loss for task τ when the kth subspace (where $k = 1, \ldots, K$) is used to construct its task model. The meta-learner updates $\{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$ by performing one gradient

Algorithm 1 MUltiple Subspaces for Meta-Learning (MUSML).

Require: stepsize α, $\{\eta_t\}$; number of inner gradient steps T_{in}, number of subspaces K, subspace dimension m, temperature $\{\gamma_t\}$; initialization $\mathbf{v}^{(0)}$;

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: sample a task τ with $\mathcal{D}_\tau^{tr}$ and $\mathcal{D}_\tau^{vl}$;
3: *base learner*:
4: **for** $k = 1, \ldots, K$ **do**
5: initialize $\mathbf{v}_{\tau,k}^{(0)} = \mathbf{v}^{(0)}$;
6: **for** $t' = 0, 1, \ldots, T_{in}-1$ **do**
7: $\mathbf{v}_{\tau,k}^{(t'+1)} = \mathbf{v}_{\tau,k}^{(t')} - \alpha\nabla_{\mathbf{v}_{\tau,k}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k,t}\mathbf{v}_{\tau,k}^{(t')})$;
8: **end for**
9: $\mathbf{v}_{\tau,k} \equiv \mathbf{v}_{\tau,k}^{(T_{in})}$;
10: $o_{\tau,k} = \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k,t}\mathbf{v}_{\tau,k})$;
11: **end for**
12: *meta-learner*:
13: $\mathcal{L}_{vl} = \sum_{k=1}^K \frac{\exp(-o_{\tau,k}/\gamma_t)}{\sum_{k'=1}^K \exp(-o_{\tau,k'}/\gamma_t)}\mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{S}_{k,t}\mathbf{v}_{\tau,k})$;
14: $\{\mathbf{S}_{1,t+1}, \ldots, \mathbf{S}_{k,t+1}\} = \{\mathbf{S}_{1,t}, \ldots, \mathbf{S}_{k,t}\} - \eta_t\nabla_{\{\mathbf{S}_{1,t},\ldots,\mathbf{S}_{k,t}\}}\mathcal{L}_{vl}$;
15: **end for**
16: **Return** $\mathbf{S}_{1,T}, \ldots, \mathbf{S}_{K,T}$.

update on the weighted validation loss (steps 13-14):

$$\sum_{k=1}^K \frac{\exp(-o_{\tau,k}/\gamma)}{\sum_{k'=1}^K \exp(-o_{\tau,k'}/\gamma)}\mathcal{L}(\mathcal{D}_\tau^{vl}; \mathbf{S}_k\mathbf{v}_{\tau,k}), \tag{2}$$

where $\gamma > 0$ is the temperature. When γ is close to 0, the softmax selection becomes one-hot; whereas when γ increases to ∞, the selection becomes uniform. In practice, we start at a high temperature and anneal to a small but nonzero temperature as in (Jang et al., 2016; Chen et al., 2020; Zhou et al., 2021b). Note that $\{o_{\tau,k} : k = 1, \ldots, K\}$ depend on the bases and $\nabla_{\{\mathbf{S}_1,\ldots,\mathbf{S}_k\}}o_{\tau,k}$ can be computed by auto-differentiation.

At meta-testing, for each testing task τ', we assign τ' to the subspace with the lowest training loss, i.e., $\mathbf{w}_{\tau'} = \mathbf{S}_{k_{\tau'}}\mathbf{v}_{\tau',k_{\tau'}}$, where $k_{\tau'} \equiv \arg\min_{1\le k\le K}\mathcal{L}(\mathcal{D}_{\tau'}^{tr}; \mathbf{S}_k\mathbf{v}_{\tau',k})$ is the chosen subspace index.

3.3. Analysis

In this section, we study the generalization performance of the learned subspace bases $\mathcal{S} \equiv \{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$ at meta-testing. The following assumptions on smoothness and compactness are standard in meta-learning (Bao et al., 2021; Grazzi et al., 2020; Fallah et al., 2020) and bilevel optimization (Franceschi et al., 2018; Bao et al., 2021). The boundedness assumption on the loss function is widely used in analyzing meta-learning algorithms (Maurer & Jaakkola,

2005; Pentina & Lampert, 2014; Amit & Meir, 2018) and traditional machine learning algorithms (Bousquet & Elisseeff, 2002).

Assumption 3.2. (i) $\ell(f(\mathbf{x};\mathbf{w}),y)$ and $\nabla_{\mathbf{w}}\ell(f(\mathbf{x};\mathbf{w}),y)$ are ϱ-Lipschitz and β-Lipschitz in $\mathbf{w}$, respectively;[1] (ii) $\{\mathbf{v}_{\tau,k}:\tau\sim p(\tau),k=1,\ldots,K\}$ and column vectors of $\mathbf{S}_k$ $(k=1,\ldots,K)$ are in a compact set, and their ℓ_2-norms are upper bounded by a constant $\rho>0$. (iii) $\ell(\cdot,\cdot)$ is upper bounded by a constant $\nu>0$.

Let $\mathcal{R}(\mathcal{S}) \equiv \mathbb{E}_{\tau'}\mathbb{E}_{\mathcal{D}_{\tau'}^{tr}}\mathbb{E}_{(\mathbf{x},y)\sim\tau'}\ell(f(\mathbf{x};\mathbf{S}_{k_{\tau'}}\mathbf{v}_{\tau',k_{\tau'}}),y)$ be the expected population risk, and $\hat{\mathcal{R}}(\mathcal{S}) \equiv \mathbb{E}_{\tau'}\mathbb{E}_{\mathcal{D}_{\tau'}^{tr}}\mathcal{L}(\mathcal{D}_{\tau'}^{tr};\mathbf{S}_{k_{\tau'}}\mathbf{v}_{\tau',k_{\tau'}})$ be the expected empirical risk averaged over all tasks. The following Theorem characterizes the generalization gap, i.e., the gap between the population and empirical risks. All proofs are in Appendix B.

Theorem 3.3. *With Assumption 3.2, we have*

$$\mathcal{R}(\mathcal{S}) \le \hat{\mathcal{R}}(\mathcal{S}) + K\sqrt{\frac{\nu^2+12\varrho\nu(1+m\alpha\delta)^{T_{in}}}{2N_{tr}}}, \quad (3)$$

where $\delta=\beta\rho^2>0$.

The proof is based on the connection between generalization and stability (Bousquet & Elisseeff, 2002). The dependence on N_{tr} in (3) agrees with their Theorem 11, as the stability constant in Algorithm 1 is of the order $\mathcal{O}(1/N_{tr})$ (Lemma B.2 in Appendix B). From (3), one can observe that increasing the subspace complexity (i.e., m or K) increases the upper bound of $\mathcal{R}(\mathcal{S})-\hat{\mathcal{R}}(\mathcal{S})$.

Next, we study the expected empirical risk $\hat{\mathcal{R}}(\mathcal{S})$, which requires the following assumption.

Assumption 3.4. (i) $\|\nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_{\tau'}^{tr};\mathbf{w})\|^2 \ge \lambda(\mathcal{L}(\mathcal{D}_{\tau'}^{tr};\mathbf{w})-L_\star)\ge 0$ for some $\lambda>0$ and $L_\star\ge 0$. (ii) $\min_{1\le k\le K}\sigma_{\min}(\mathbf{S}_k) \ge c_1(1-c_2\sqrt{m/d})$, where $c_1,c_2>0$.

The first assumption is called the Polyak-Łojasiewicz condition (Polyak, 1963; Lojasiewicz, 1963), and has been used in the non-convex optimization literature (Karimi et al., 2016; Liu et al., 2022). The second assumption on the smallest singular value is a property of random matrices with high probability (Rudelson & Vershynin, 2009).

Theorem 3.5. *With Assumptions 3.2 and 3.4, if* $\alpha<\min(\frac{1}{m\beta\rho^2},\frac{1}{\lambda},\frac{2}{\lambda c_1^2})$*, we have*

$$\hat{\mathcal{R}}(\mathcal{S}) \le \sqrt{4L_\star^2+2\nu^2\big((\sqrt{m}-\kappa_1)(\sqrt{m}+\kappa_2)\phi\big)^{2T_{in}}} + \frac{\nu}{K+\nu}, \quad (4)$$

where $\kappa_1=(\sqrt{d}/c_2)(\sqrt{2/(c_1^2\alpha\lambda)}+1)$, $\kappa_2=(\sqrt{d}/c_2)(\sqrt{2/(c_1^2\alpha\lambda)}-1)$*, and* $\phi=c_1c_2\sqrt{\alpha\lambda/(2d)}$ *are all positive.*

The above Theorem shows that increasing K reduces the upper bound of $\hat{\mathcal{R}}(\mathcal{S})$. This is intuitive as we choose the subspace with the lowest training loss during meta-testing.

Combining Theorems 3.3 and 3.5, we obtain the following Corollary. It indicates that the (upper bound on) population risk $\mathcal{R}(\mathcal{S})$ may not always decrease as K increases, due to overfitting.

Corollary 3.6. *With Assumptions 3.2 and 3.4, if* $\alpha<\min(\frac{1}{m\beta\rho^2},\frac{1}{\lambda},\frac{2}{\lambda c_1^2})$*, we have*

$$\mathcal{R}(\mathcal{S}) \le K\sqrt{\frac{\nu^2+12\varrho\nu(1+m\alpha\delta)^{T_{in}}}{2N_{tr}}}+\frac{\nu}{K+\nu} + \sqrt{4L_\star^2+2\nu^2\big((\sqrt{m}-\kappa_1)(\sqrt{m}+\kappa_2)\phi\big)^{2T_{in}}}. \quad (5)$$

Let $\mathbf{w}_{\tau'}^\star \equiv \arg\min_{\mathbf{w}_{\tau'}}\mathbb{E}_{(\mathbf{x},y)\sim\tau'}\ell(f(\mathbf{x};\mathbf{w}_{\tau'}),y)$ be the optimal task model for task τ', and $\mathcal{R}^\star \equiv \mathbb{E}_{\tau'}\mathbb{E}_{(\mathbf{x},y)\sim\tau'}\ell(f(\mathbf{x};\mathbf{w}_{\tau'}^\star),y)$ be the minimum expected loss averaged over all tasks. The following Theorem provides an upper bound on the expected excess risk (Zhou et al., 2021a) $\mathcal{R}(\mathcal{S})-\mathcal{R}^\star$, which compares the performance of the learned task model with that of the optimal model.

Theorem 3.7. *With Assumption 3.2, we have*

$$\mathcal{R}(\mathcal{S})-\mathcal{R}^\star \le \rho\sqrt{m}\,\mathbb{E}_{\tau'}\mathbb{E}_{\mathcal{D}_{\tau'}^{tr}}\|\mathbf{v}_{\tau',k_{\tau'}}-\mathbf{v}_{\tau',k_{\tau'}}^\star\| + \varrho\mathbb{E}_{\tau'}\mathbb{E}_{\mathcal{D}_{\tau'}^{tr}}\mathrm{dist}(\mathbf{w}_{\tau'}^\star,\mathbb{S}_{k_{\tau'}}) + K\sqrt{\frac{\nu^2+12\varrho\nu(1+m\alpha\delta)^{T_{in}}}{2N_{tr}}},$$

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

From Theorem 3.7, $\mathcal{R}(\mathcal{S})-\mathcal{R}^\star$ is upper-bounded by three terms: (i) The first term measures the distance between the approximate minimizer $\mathbf{v}_{\tau',k_{\tau'}}$ and exact minimizer $\mathbf{v}_{\tau',k_{\tau'}}^\star$; (ii) The second term arises from the approximation error of $\mathbf{w}_{\tau'}^\star$ using the learned subspaces; (iii) The third term depends on the complexity of subspaces (i.e., m and K). For the centroid-based clustering method in (Zhou et al., 2021a), the upper bound of its expected excess risk contains a term $\mathbb{E}_{\tau'}\mathbb{E}_{\mathcal{D}_{\tau'}^{tr}}\|\boldsymbol{\omega}_{k_{\tau'}^\star}-\mathbf{w}_{\tau'}^\star\|^2$, where $\boldsymbol{\omega}_{k_{\tau'}^\star}$ is the centroid of the cluster that τ' is assigned to. The distance $\|\boldsymbol{\omega}_{k_{\tau'}^\star}-\mathbf{w}_{\tau'}^\star\|^2$ plays the same role as the term $\mathrm{dist}(\mathbf{w}_{\tau'}^\star,\mathbb{S}_{k_{\tau'}})$ in Theorem 3.7, which measures how far the optimal model $\mathbf{w}_{\tau'}^\star$ is away from the subspaces or clusters.

4. Experiments

In this section, we perform extensive regression and classification experiments to demonstrate effectiveness of the

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

proposed method.

4.1. Few-shot Regression on Synthetic Data

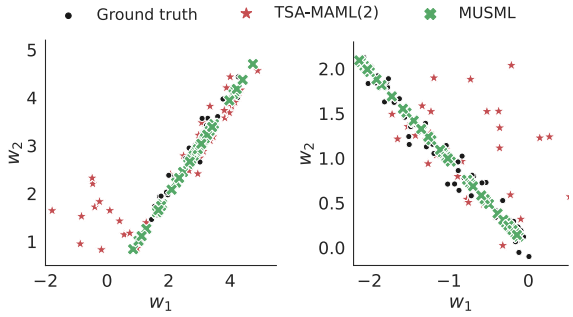
In this experiment, we use a synthetic 1-dimensional data set to examine whether MUSML can discover subspaces that the task model parameters lie in. We use a 5-shot regression setting, with $14,000$ meta-training, $2,000$ meta-validation, and $6,000$ meta-testing tasks. The model for task τ is $f(x; \mathbf{w}_\tau) = \exp(0.1 w_{\tau,1} x) + w_{\tau,2} |\sin(x)|$, in which $\mathbf{w}_\tau = [w_{\tau,1}; w_{\tau,2}]$ is randomly sampled from one of the two subspaces: (i) *Line-A*: $\mathbf{w}_\tau = \mathbf{S}_1 a_\tau + 0.1\boldsymbol{\xi}_\tau$, where $\mathbf{S}_1 = [1; 1]$, $a_\tau \sim \mathcal{U}(1, 5)$, and $\boldsymbol{\xi}_\tau \sim \mathcal{N}(\mathbf{0}; \mathbf{I})$; and (ii) *Line-B*: $\mathbf{w}_\tau = \mathbf{S}_2 a_\tau + 0.1\boldsymbol{\xi}_\tau$, where $\mathbf{S}_2 = [-1; 1]$, $a_\tau \sim \mathcal{U}(0, 2)$, and $\boldsymbol{\xi}_\tau \sim \mathcal{N}(\mathbf{0}; \mathbf{I})$. The samples of task τ are generated as $y = f(x; \mathbf{w}_\tau) + 0.05\xi$, where $x \sim \mathcal{U}(-5, 5)$ and $\xi \sim \mathcal{N}(0, 1)$. The experiment is repeated 10 times with different seeds. Implementation details are in Appendix A.1.1.

The proposed MUSML (with $K = 2, m = 1$) is compared with the following meta-learning baselines: (i) MAML (Finn et al., 2017), (ii) BMG (Flennerhag et al., 2022), which uses target bootstrap, and structured meta-learning algorithms including (iii) Dirichlet process mixture model (DPMM) (Jerfel et al., 2019), (iv) hierarchically structured meta-learning (HSML) (Yao et al., 2019), (v) automated relational meta-learning (ARML) (Yao et al., 2020) using a graph structure, and (vi) task similarity aware meta-learning (TSA-MAML) (Zhou et al., 2021a) with different numbers of clusters. We use these baselines' official implementations (except for DPMM and BMG whose implementations are not publicly available). For performance evaluation, the mean squared error (MSE) on the meta-testing set is used.

Results. Table 1 shows the meta-testing MSE. As can be seen, structured meta-learning methods (DPMM, HSML, ARML, TSA-MAML, and MUSML) are significantly better than methods with a globally-shared meta-model (MAML and BMG). In particular, MUSML performs the best. Furthermore, simply increasing the number of clusters in TSA-MAML fails to beat MUSML.

Table 1. Meta-testing MSE (with standard deviation) of 5-shot regression on synthetic data. For TSA-MAML, the number in brackets is the number of clusters used.

MAML (Finn et al., 2017)	0.74 ± 0.03
BMG (Flennerhag et al., 2022)	0.67 ± 0.03
DPMM (Jerfel et al., 2019)	0.56 ± 0.09
HSML (Yao et al., 2019)	0.49 ± 0.10
ARML (Yao et al., 2020)	0.60 ± 0.07
TSA-MAML(2) (Zhou et al., 2021a)	0.58 ± 0.10
TSA-MAML(10) (Zhou et al., 2021a)	0.24 ± 0.09
TSA-MAML(20) (Zhou et al., 2021a)	0.12 ± 0.10
TSA-MAML(40) (Zhou et al., 2021a)	0.14 ± 0.09
TSA-MAML(80) (Zhou et al., 2021a)	0.13 ± 0.08
MUSML	$\mathbf{0.07} \pm 0.01$



(a) Tasks from *Line-A*. (b) Tasks from *Line-B*.

Figure 1. Visualization of task model parameters.

Figure 1 visualizes the task model parameters obtained by TSA-MAML(2) and MUSML on 100 randomly sampled meta-testing tasks (50 per subspace). As can be seen, MUSML successfully discovers the underlying subspaces, while the centroid-based clustering method TSA-MAML does not. Table 11 of Appendix A.1.2 also shows that MUSML is more accurate in estimating the task model parameters.

4.2. Few-shot Regression on *Pose* Data

While the synthetic data used in the previous experiment is tailored for the proposed subspace model, in this section, we perform experiments on a real-world pose prediction dataset from (Yin et al., 2020). This is created based on the Pascal 3D data (Xiang et al., 2014). Each object contains 100 samples, where input $\mathbf{x}$ is a 128×128 grey-scale image and output y is its orientation relative to a fixed canonical pose. Following (Yin et al., 2020), we adopt a 15-shot regression setting and randomly select 50 objects for meta-training, 15 for meta-validation, and 15 for meta-testing. The experiment is repeated 15 times with different seeds.

The MR-MAML regularization (Yin et al., 2020) is used on all the methods except the vanilla MAML. MUSML uses the same encoder-decoder network in (Yin et al., 2020) as the model $f(\mathbf{x}; \mathbf{w})$. Hyperparameters K and m, as well as the number of clusters in TSA-MAML, are chosen from 1 to 5 using the meta-validation set. For performance evaluation, the MSE on the meta-testing set is used.

Table 2 shows the meta-testing MSE. As can be seen, MUSML is again better than the other baselines, confirming the effectiveness of the learned subspaces.

4.3. Few-shot Classification

Setup. In this experiment, we use three meta-datasets: (i) *Meta-Dataset-BTAF*, proposed in (Yao et al., 2019), which consists of four image classification datasets: (a) *Bird*; (b) *Texture*; (c) *Aircraft*; and (d) *Fungi*. Sample images are shown in Figure 2. (ii) *Meta-Dataset-ABF*, proposed in (Zhou et al., 2021a), which consists of *Aircraft*, *Bird*,

Table 2. Meta-testing MSE (with standard deviation) of 15-shot regression on *Pose*. Results on MAML and MR-MAML are from (Yin et al., 2020).

MAML (Finn et al., 2017)	5.39 ± 1.31
MR-MAML (Yin et al., 2020)	2.26 ± 0.09
BMG (Flennerhag et al., 2022)	2.16 ± 0.15
DPMM (Jerfel et al., 2019)	1.99 ± 0.08
HSML (Yao et al., 2019)	2.04 ± 0.13
ARML (Yao et al., 2020)	2.21 ± 0.15
TSA-MAML (Zhou et al., 2021a)	1.96 ± 0.07
MUSML	$\mathbf{1.83} \pm 0.05$

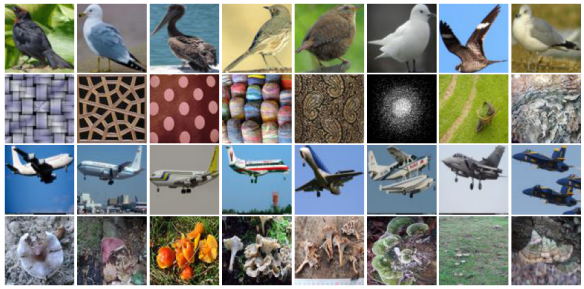


Figure 2. Some random images from the meta-testing set of *Meta-Dataset-BTAF* (Top to bottom: *Bird*, *Texture*, *Aircraft*, and *Fungi*).

and *Fungi*. (iii) *Meta-Dataset-CIO*, which consists of three widely-used few-shot datasets: *CIFAR-FS* (Bertinetto et al., 2018), *mini-ImageNet* (Vinyals et al., 2016), and *Omniglot* (Lake et al., 2015). We use the meta-training/meta-validation/meta-testing splits in (Yao et al., 2020; Zhou et al., 2021a; Lake et al., 2015). A summary of the datasets is in Table 3.

As for the network architecture, we use the standard *Conv4* backbone (Finn et al., 2017; Yao et al., 2020; Zhou et al., 2021a), and a simple prototype classifier with cosine similarity on top (Snell et al., 2017; Gidaris & Komodakis, 2018) as $f(\mathbf{x}; \mathbf{w})$. Hyperparameters K and m are chosen from 1 to 5 on the meta-validation set. Implementation details are in Appendix A.2.

MUSML is compared with the following state-of-the-arts in the 5-way 5-shot and 5-way 1-shot settings: (i) meta-learning algorithms with a globally-shared meta-model including MAML, ProtoNet, ANIL (Raghu et al., 2020), and BMG; (ii) structured meta-learning algorithms including DPMM, HSML, ARML, TSA-MAML and its variant using ProtoNet as the base learner (denoted TSA-ProtoNet). The number of clusters in TSA-MAML and TSA-ProtoNet are tuned from 1 to 5 on the meta-validation set. For performance evaluation, the classification accuracy on the meta-testing set is used. The experiment is repeated 5 times with different seeds.

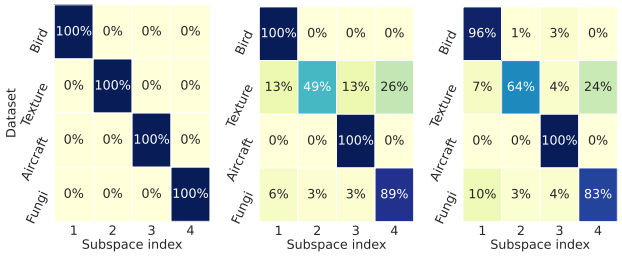
Table 3. Statistics of the datasets.

	#classes (meta-training/meta-validation/meta-testing)
Bird	64/16/20
Texture	30/7/10
Aircraft	64/16/20
Fungi	64/16/20
CIFAR-FS	64/16/20
mini-ImageNet	64/16/20
Omniglot	71/15/16

4.3.1. *Meta-Dataset-BTAF*

Table 4 shows the 5-shot results. As can be seen, MUSML is more accurate than both structured and unstructured meta-learning methods, demonstrating the benefit of structuring task model parameters into subspaces. Figure 3 shows the assignment of tasks to the learned subspaces in MUSML. As can be seen, meta-training tasks from the same dataset are always assigned to the same subspace, demonstrating that MUSML can discover the task structure from meta-training tasks. Though the meta-validation and meta-testing classes are not seen during meta-training, most of the corresponding tasks are still assigned to the correct subspaces. The assignment for *Texture* is slightly worse, as the *Texture* and *Fungi* images are more similar to each other (Figure 2).

Table 5 shows the 1-shot results. MUSML, while still the best overall, has a smaller improvement than in the 5-shot setting. This suggests that having more training samples is beneficial for the base learner to choose a proper subspace. The assignment of tasks to the learned subspaces is shown in Figure 9 of Appendix A.3.



(a) Meta-training. (b) Meta-validation. (c) Meta-testing.

Figure 3. Task assignment to the learned subspaces in 5-way 5-shot setting on *Meta-Dataset-BTAF* (the number of subspaces K selected by the meta-validation set is 4). Darker color indicates higher percentage.

4.3.2. *Meta-Dataset-ABF* AND *Meta-Dataset-CIO*

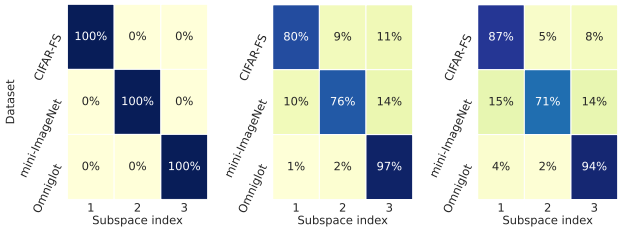
Tables 6 and 7 show the results on *Meta-Dataset-ABF* and *Meta-Dataset-CIO*, respectively. Here, we only consider the 5-shot setting, which is more useful for subspace learning. As can be seen, MUSML consistently outperforms centroid-based clustering methods (DPMM, TSA-

Table 4. 5-way 5-shot accuracy (with 95% confidence interval) on *Meta-Dataset-BTAF*. Results marked with † are from (Yao et al., 2020).

	Bird	*Texture*	*Aircraft*	*Fungi*	average
MAML† (Finn et al., 2017)	68.52 ± 0.73	44.56 ± 0.68	66.18 ± 0.71	51.85 ± 0.85	57.78
ProtoNet (Snell et al., 2017)	71.48 ± 0.72	50.36 ± 0.67	71.67 ± 0.69	55.68 ± 0.82	62.29
ANIL (Raghu et al., 2020)	70.67 ± 0.72	44.67 ± 0.95	66.05 ± 1.07	52.89 ± 0.30	58.57
BMG (Flennerhag et al., 2022)	71.56 ± 0.76	49.44 ± 0.73	66.83 ± 0.79	52.56 ± 0.89	60.10
DPMM (Jerfel et al., 2019)	72.22 ± 0.70	49.32 ± 0.68	73.55 ± 0.69	56.82 ± 0.81	63.00
TSA-MAML (Zhou et al., 2021a)	72.31 ± 0.71	49.50 ± 0.68	74.01 ± 0.70	56.95 ± 0.80	63.20
HSML† (Yao et al., 2019)	71.68 ± 0.73	48.08 ± 0.69	73.49 ± 0.68	56.32 ± 0.80	62.39
ARML† (Yao et al., 2020)	73.68 ± 0.70	49.67 ± 0.67	74.88 ± 0.64	57.55 ± 0.82	63.95
TSA-ProtoNet (Zhou et al., 2021a)	73.70 ± 0.73	50.91 ± 0.74	73.55 ± 0.78	56.11 ± 0.82	63.57
MUSML	$\mathbf{76.79} \pm 0.72$	$\mathbf{52.41} \pm 0.75$	$\mathbf{77.76} \pm 0.82$	$\mathbf{57.74} \pm 0.81$	**66.18**

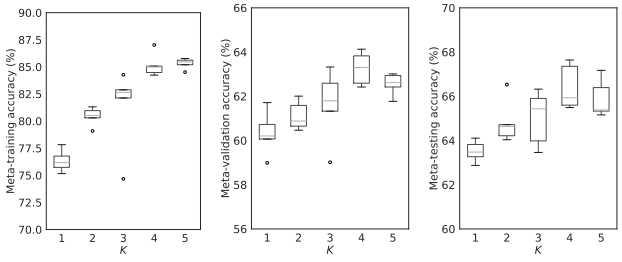
Table 5. 5-way 1-shot accuracy (with 95% confidence interval) on *Meta-Dataset-BTAF*. Results marked with † are from (Yao et al., 2020).

	Bird	*Texture*	*Aircraft*	*Fungi*	average
MAML† (Finn et al., 2017)	53.94 ± 1.45	31.66 ± 1.31	51.37 ± 1.38	42.12 ± 1.36	44.77
ProtoNet (Snell et al., 2017)	60.37 ± 1.31	40.57 ± 0.78	52.83 ± 0.93	44.10 ± 1.36	49.50
ANIL (Raghu et al., 2020)	53.36 ± 1.42	31.91 ± 1.25	52.87 ± 1.34	42.30 ± 1.28	45.11
BMG (Flennerhag et al., 2022)	54.12 ± 1.46	32.19 ± 1.21	52.09 ± 1.35	43.00 ± 1.37	45.35
DPMM (Jerfel et al., 2019)	61.30 ± 1.47	35.21 ± 1.35	57.88 ± 1.37	43.81 ± 1.45	49.55
TSA-MAML (Zhou et al., 2021a)	61.37 ± 1.42	35.41 ± 1.39	$\mathbf{58.78} \pm 1.37$	44.17 ± 1.25	49.93
HSML† (Yao et al., 2019)	60.98 ± 1.50	35.01 ± 1.36	57.38 ± 1.40	44.02 ± 1.39	49.35
ARML† (Yao et al., 2020)	$\mathbf{62.33} \pm 1.47$	35.65 ± 1.40	58.56 ± 1.41	44.82 ± 1.38	50.34
TSA-ProtoNet (Zhou et al., 2021a)	60.41 ± 1.02	40.98 ± 1.20	53.29 ± 0.89	43.91 ± 1.31	49.64
MUSML	60.52 ± 0.33	$\mathbf{41.33} \pm 1.30$	54.69 ± 0.69	$\mathbf{45.60} \pm 0.43$	**50.53**



(a) Meta-training. (b) Meta-validation. (c) Meta-testing.

Figure 4. Task assignment to the learned subspaces in 5-way 5-shot setting on *Meta-Dataset-CIO* (K selected by the meta-validation set is 3). Darker color indicates higher percentage.



(a) Meta-training. (b) Meta-validation. (c) Meta-testing.

Figure 5. 5-way 5-shot classification accuracy on *Meta-Dataset-BTAF* with varying K (m is fixed at 2).

MAML, TSA-ProtoNet) and structured meta-learning methods (HSML, ARML). MUSML again outperforms methods with a globally-shared meta-model (MAML, ProtoNet, ANIL, BMG), confirming the effectiveness of using a subspace mixture. The performance of MUSML on *Omniglot* is slightly worse in Table 7. This may be due to that *Omniglot* is a simple dataset and a single meta-model is good enough. As shown in Figure 4, its meta-validation and meta-testing tasks are often assigned to the same subspace.

4.3.3. Effects of K and m

In this experiment, we study the effects of K and m on the 5-shot performance of MUSML on *Meta-Dataset-BTAF*.

Figure 5(a) shows that the meta-training accuracy increases with K. However, a large $K = 5$ is not advantageous at meta-validation (Figure 5(b)) and meta-testing (Figure 5(c)).

Figures 6(b) and 6(c) show that the meta-validation and meta-testing accuracies of MUSML increase when m increases from 1 to 2, but larger m's ($m = 3, 4, 5$) lead to worse performance. This is because the obtained task model parameters ($\mathbf{W}$) lie close to the union of 2-dimensional subspaces[2], and so a larger m does not improve performance.

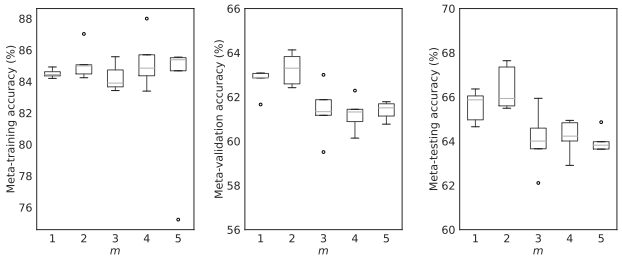
[2]For example, for the $\mathbf{W}$ solution obtained with $m = 5$ on *Meta-Dataset-BTAF* (under 5-way 5-shot setting), approximation by a rank-2 matrix $\hat{\mathbf{W}}$ leads to a relative error ($\|\mathbf{W} - \hat{\mathbf{W}}\|_F / \|\mathbf{W}\|_F$) of only 4.1%.

Table 6. Accuracy (with 95% confidence interval) of 5-way 5-shot classification on *Meta-Dataset-ABF*. Results marked with † are from (Zhou et al., 2021a).

	Aircraft	*Bird*	*Fungi*	average
MAML† (Finn et al., 2017)	67.82 ± 0.65	70.55 ± 0.77	53.20 ± 0.82	63.86
ProtoNet(Snell et al., 2017)	69.74 ± 0.64	71.46 ± 0.69	55.66 ± 0.68	65.62
ANIL (Raghu et al., 2020)	69.24 ± 0.87	70.34 ± 1.20	53.71 ± 0.67	64.43
BMG (Flennerhag et al., 2022)	69.75 ± 0.72	73.04 ± 0.77	54.61 ± 0.84	65.80
DPMM (Jerfel et al., 2019)	70.22 ± 0.69	73.28 ± 1.33	54.28 ± 1.01	66.26
TSA-MAML† (Zhou et al., 2021a)	72.84 ± 0.63	74.80 ± 0.76	56.86 ± 0.67	68.17
HSML† (Yao et al., 2019)	69.89 ± 0.90	68.99 ± 1.01	53.63 ± 1.03	64.17
ARML (Yao et al., 2020)	70.20 ± 0.91	69.12 ± 1.01	54.23 ± 1.07	64.52
TSA-ProtoNet (Zhou et al., 2021a)	74.42 ± 0.62	75.11 ± 0.72	56.77 ± 0.69	68.77
MUSML	$\mathbf{79.88} \pm 0.61$	$\mathbf{75.63} \pm 0.73$	$\mathbf{57.80} \pm 0.80$	**71.10**

Table 7. Accuracy (with 95% confidence interval) of 5-way 5-shot classification on *Meta-Dataset-CIO*.

	CIFAR-FS	*mini-ImageNet*	*Omniglot*	average
MAML (Finn et al., 2017)	66.28 ± 1.61	60.20 ± 1.20	96.91 ± 0.39	74.46
ProtoNet (Snell et al., 2017)	71.32 ± 1.54	62.90 ± 1.07	95.32 ± 0.25	76.51
ANIL (Raghu et al., 2020)	66.08 ± 0.90	60.62 ± 0.94	97.13 ± 0.13	74.61
BMG (Flennerhag et al., 2022)	70.49 ± 1.22	63.97 ± 1.19	$\mathbf{97.92} \pm 0.42$	77.46
DPMM (Jerfel et al., 2019)	69.84 ± 1.42	62.92 ± 1.28	97.14 ± 0.28	76.63
TSA-MAML (Zhou et al., 2021a)	71.11 ± 1.55	62.57 ± 1.31	96.99 ± 0.31	76.89
HSML (Zhou et al., 2019)	69.24 ± 1.57	62.28 ± 1.23	95.10 ± 0.32	75.54
ARML (Yao et al., 2020)	68.88 ± 1.91	63.26 ± 1.33	96.23 ± 0.31	76.12
TSA-ProtoNet (Zhou et al., 2021a)	72.37 ± 1.46	63.23 ± 1.52	96.21 ± 0.33	77.27
MUSML	$\mathbf{73.25} \pm 1.42$	$\mathbf{65.12} \pm 1.48$	95.13 ± 0.28	**77.83**



(a) Meta-training. (b) Meta-validation. (c) Meta-testing.

Figure 6. 5-way 5-shot classification accuracy on *Meta-Dataset-BTAF* with varying m (K is fixed at 4).

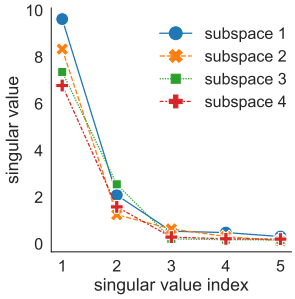


Figure 7. Singular values of model parameters of meta-testing tasks under the 5-way 5-shot setting on *Meta-Dataset-BTAF* ($K = 4$ and $m = 5$).

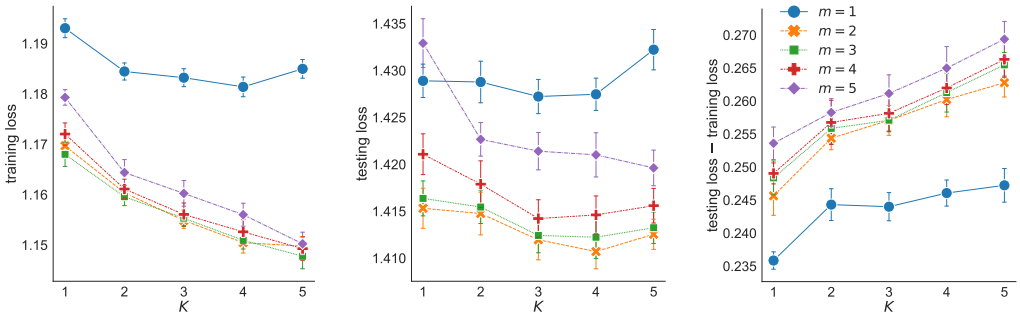
Figure 7 also shows that for the 4 subspaces, the first 2 singular values of $\mathbf{W}$ are dominant.

To demonstrate the theoretical results in Section 3.3, we further study the effects of K and m on the meta-testing loss. The average training (resp. testing) loss of meta-testing tasks is an empirical estimate of $\hat{\mathcal{R}}(\mathcal{S})$ (resp. $\mathcal{R}(\mathcal{S})$), while their gap measures the generalization performance.

Figure 8(a) shows that, for $m \geq 2$, increasing K leads to a reduction in the training loss, which is suggested by Theorem 3.5. However, the testing loss does not always decrease when K increases (Figure 8(b)), which also agrees with Corollary 3.6. Figure 8(c) shows that a large K or m may enlarge the generalization gap, which justifies Theorem 3.3. As shown in Figure 8(c), the generalization gap is approximately linear with K, which agrees with the relationship between the upper bound of $\mathcal{R}(\mathcal{S}) - \hat{\mathcal{R}}(\mathcal{S})$ and K in Theorem 3.3.

4.4. Cross-Domain Few-Shot Classification

We examine the effectiveness of MUSML on cross-domain few-shot classification, which is more challenging as the



(a) Training loss. (b) Testing loss. (c) Generalization gap.

Figure 8. Effects of K and m on the training loss, testing loss, and generalization gap (with 95% confidence interval) of meta-testing tasks under the 5-way 5-shot setting on *Meta-Dataset-BTAF*.

Table 8. Accuracy of cross-domain 5-way 5-shot classification (*Meta-Dataset-BTAF* $\rightarrow$ *Meta-Dataset-CIO*).

MAML	ProtoNet	ANIL	BMG	DPMM	TSA-MAML	HSML	ARML	TSA-ProtoNet	MUSML
64.25	66.13%	65.19%	66.98%	66.73%	66.85%	65.18%	65.37%	66.92%	**67.41%**

Table 9. Accuracy of 5-way 5-shot classification on *Meta-Dataset-BTAF*.

γ	0.0001	0.001	0.01	0.1	1.0	2.0	MUSML
accuracy	51.22%	60.12%	61.15%	63.16%	62.11%	62.02%	**66.18%**

Table 10. Accuracy of 5-way 5-shot classification on meta-datasets.

	Meta-Dataset-BTAF	*Meta-Dataset-ABF*	*Meta-Dataset-CIO*
Meta-Curvature (Park & Oliva, 2019)	60.02%	64.51%	76.13%
MUSML-Curvature	66.10%	69.23%	77.96%
Meta-SGD (Li et al., 2017)	58.93%	64.19%	75.95%
MUSML-SGD	65.72%	69.15%	77.48%

testing domain is unseen at meta-training. We perform 5-way 5-shot classification, where *Meta-Dataset-BTAF* is used for meta-training, and *Meta-Dataset-CIO* for meta-testing. Table 8 shows the meta-testing accuracy. As can be seen, MUSML is also effective on unseen domains.

4.5. Effects of Temperature Scaling Schedule

The temperature schedule used is linear annealing as in DynamicConvolution (Chen et al., 2020) and ProbMask (Zhou et al., 2021b). We conduct a 5-way 5-shot experiment on *Meta-Dataset-BTAF* to evaluate MUSML with a constant temperature. Table 9 reports the meta-testing accuracy. We can see that using a constant γ is inferior.

4.6. Improving Existing Meta-Learning Approaches

As the proposed MUSML is general, a subspace mixture is also useful for other meta-learning approaches. In this experiment, we combine MUSML with Meta-Curvature (Park & Oliva, 2019) and Meta-SGD (Li et al., 2017). Table 10 reports 5-way 5-shot accuracies on meta-datasets. As can be seen, MUSML is beneficial for both Meta-Curvature and Meta-SGD.

5. Conclusion

In this paper, we formulate task model parameters into a subspace mixture and propose a model-agnostic meta-learning algorithm with subspace learning called MUSML. For each task, the base learner builds a task model from each subspace, while the meta-learner updates the meta-parameters by minimizing a weighted validation loss. The generalization performance is theoretically studied. Experimental results on benchmark datasets for classification and regression validate the effectiveness of the proposed MUSML.

Acknowledgements

This work was supported by NSFC key grant 62136005, NSFC general grant 62076118, and Shenzhen fundamental research program JCYJ20210324105000003. This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grant 16200021).

References

Amit, R. and Meir, R. Meta-learning by adjusting priors based on extended PAC-Bayes theory. In *International Conference on Machine Learning*, pp. 205–214, 2018.

Bao, F., Wu, G., Li, C., Zhu, J., and Zhang, B. Stability and generalization of bilevel programming in hyperparameter optimization. In *Neural Information Processing Systems*, 2021.

Bengio, Y., Bengio, S., and Cloutier, J. Learning a synaptic learning rule. In *International Joint Conference on Neural Networks*, pp. 969–975, 1991.

Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2018.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., and Liu, Z. Dynamic convolution: Attention over convolution kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11030–11039, 2020.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Learning to learn around a common mean. In *Neural Information Processing Systems*, pp. 10169–10179, 2018.

Denevi, G., Ciliberto, C., Grazzi, R., and Pontil, M. Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning*, pp. 1566–1575, 2019.

Denevi, G., Pontil, M., and Ciliberto, C. The advantage of conditional meta-learning for biased regularization and fine tuning. In *Neural Information Processing Systems*, 2020.

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

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Flennerhag, S., Schroecker, Y., Zahavy, T., van Hasselt, H., Silver, D., and Singh, S. Bootstrapped meta-learning. In *International Conference on Learning Representations*, 2022.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pp. 1568–1577, 2018.

Gidaris, S. and Komodakis, N. Dynamic few-shot visual learning without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

Grazzi, R., Franceschi, L., Pontil, M., and Salzo, S. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pp. 3748–3758, 2020.

Gressmann, F., Eaton-Rosen, Z., and Luschi, C. Improving neural network training in low dimensional random bases. *Neural Information Processing Systems*, 2020.

Gu, J., Wang, Y., Chen, Y., Li, V. O., and Cho, K. Meta-learning for low-resource neural machine translation. In *Empirical Methods in Natural Language Processing*, pp. 3622–3631, 2018.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. Subspace inference for Bayesian deep learning. In *Uncertainty in Artificial Intelligence*, pp. 1169–1179, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2016.

Jerfel, G., Grant, E., Griffiths, T., and Heller, K. A. Reconciling meta-learning and continual learning with online mixtures of tasks. In *Neural Information Processing Systems*, pp. 9122–9133, 2019.

Jiang, W., Kwok, J., and Zhang, Y. Effective meta-regularization by kernelized proximal regularization. *Neural Information Processing Systems*, 2021.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kong, W., Somani, R., Song, Z., Kakade, S., and Oh, S. Meta-learning for mixed linear regression. In *International Conference on Machine Learning*, pp. 5394–5404, 2020.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-SGD: Learning to learn quickly for few-shot learning. Preprint arXiv:1707.09835, 2017.

Liu, C., Zhu, L., and Belkin, M. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 2022.

Liu, H., Simonyan, K., and Yang, Y. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2018.

Lojasiewicz, S. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117(87-89):2, 1963.

Maclaurin, D., Duvenaud, D., and Adams, R. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pp. 2113–2122, 2015.

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

Munkhdalai, T. and Yu, H. Meta networks. In *International Conference on Machine Learning*, pp. 2554–2563, 2017.

Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2018.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. Preprint arXiv:1803.02999, 2018.

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

Oreshkin, B., López, P. R., and Lacoste, A. TADAM: Task dependent adaptive metric for improved few-shot learning. In *Neural Information Processing Systems*, pp. 721–731, 2018.

Park, E. and Oliva, J. B. Meta-Curvature. *Neural Information Processing Systems*, 32, 2019.

Pentina, A. and Lampert, C. A PAC-Bayesian bound for lifelong learning. In *International Conference on Machine Learning*, pp. 991–999, 2014.

Polyak, B. Gradient methods for the minimisation of functionals. *Ussr Computational Mathematics and Mathematical Physics*, 3:864–878, 1963.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? Towards understanding the effectiveness of MAML. In *International Conference on Learning Representations*, 2020.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. In *Neural Information Processing Systems*, pp. 113–124, 2019.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, pp. 5331–5340, 2019.

Rudelson, M. and Vershynin, R. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pp. 1842–1850, 2016.

Saunshi, N., Zhang, Y., Khodak, M., and Arora, S. A sample complexity separation between non-convex and convex meta-learning. In *International Conference on Machine Learning*, pp. 8512–8521, 2020.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Neural Information Processing Systems*, pp. 4077–4087, 2017.

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

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., and Larochelle, H. Meta-Dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

Tripuraneni, N., Jin, C., and Jordan, M. Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pp. 10434–10443, 2021.

Vinyals, O., Blundell, C., Lillicrap, T., and Wierstra, D. Matching networks for one shot learning. In *Neural Information Processing Systems*, pp. 3630–3638, 2016.

Vuorio, R., Sun, S.-H., Hu, H., and Lim, J. J. Multimodal model-agnostic meta-learning via task-aware modulation. In *Neural Information Processing Systems*, pp. 1–12, 2019.

Wang, R., Demiris, Y., and Ciliberto, C. Structured prediction for conditional meta-learning. *Neural Information Processing Systems*, 33:2587–2598, 2020a.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):1–34, 2020b.

Wortsman, M., Horton, M., Guestrin, C., Farhadi, A., and Rastegari, M. Learning neural network subspaces. In *International Conference on Machine Learning*, 2021.

Xiang, Y., Mottaghi, R., and Savarese, S. Beyond pascal: A benchmark for 3D object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pp. 75–82, 2014.

Yao, H., Wei, Y., Huang, J., and Li, Z. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, pp. 7045–7054, 2019.

Yao, H., Wu, X., Tao, Z., Li, Y., Ding, B., Li, R., and Li, Z. Automated relational meta-learning. In *International Conference on Learning Representations*, 2020.

Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. Meta-learning without memorization. In *International Conference on Learning Representations*, 2020.

Zhou, P., Yuan, X., Xu, H., Yan, S., and Feng, J. Efficient meta learning via minibatch proximal update. In *Neural Information Processing Systems*, pp. 1534–1544, 2019.

Zhou, P., Zou, Y., Yuan, X., Feng, J., Xiong, C., and Hoi, S. Task similarity aware meta learning: Theory-inspired improvement on MAML. In *Conference on Uncertainty in Artificial Intelligence*, 2021a.

Zhou, X., Zhang, W., Xu, H., and Zhang, T. Effective sparsification of neural networks with global sparsity constraint. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3599–3608, 2021b.

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

A. Experiment Details and Additional Results

A.1. Few-shot Regression on Synthetic Data

A.1.1. Implementation Details

The subspace bases are trained for $T = 30,000$ iterations using the Adam optimizer (Kingma & Ba, 2015). For the meta-learner, the initial learning rate is 0.001, which is then reduced by half every $5,000$ iterations. The base learner uses a learning rate of $\alpha = 0.05$, $\mathbf{v}^{(0)} = \frac{1}{m}\mathbf{1}$, and T_{in} is 5 (resp. 20) at meta-training (resp. meta-testing). The temperature is $\gamma_t = \max(10^{-5}, 0.5 - t/T)$, a linear annealing schedule as in (Chen et al., 2020; Zhou et al., 2021b). To prevent overfitting, we evaluate the meta-validation performance every $2,000$ iterations, and stop training when the meta-validation accuracy has no significant improvement for 5 consecutive evaluations.

A.1.2. Additional Results

Table 11 shows the average Euclidean distance between the estimated task model parameters and the ground truth. As can be seen, MUSML is more accurate in estimating the task models, confirming the effectiveness of the learned subspaces.

Table 11. Average Euclidean distance (with standard deviation) between the estimated task model parameters and ground-truth in 5-shot setting on synthetic data. For TSA-MAML, the number in brackets is the number of clusters used.

MAML (Finn et al., 2017)	1.69 ± 0.02
BMG (Flennerhag et al., 2022)	1.55 ± 0.03
DPMM (Jerfel et al., 2019)	0.85 ± 0.10
HSML (Yao et al., 2019)	0.80 ± 0.09
ARML (Yao et al., 2020)	0.91 ± 0.11
TSA-MAML(2) (Zhou et al., 2021a)	0.88 ± 0.12
TSA-MAML(10) (Zhou et al., 2021a)	0.47 ± 0.19
TSA-MAML(20) (Zhou et al., 2021a)	0.33 ± 0.18
TSA-MAML(40) (Zhou et al., 2021a)	0.36 ± 0.19
TSA-MAML(80) (Zhou et al., 2021a)	0.36 ± 0.18
MUSML	$\mathbf{0.17} \pm 0.01$

A.2. Few-shot Classification

We use the cross-entropy loss for $\ell(\cdot, \cdot)$. The number of parameters in Conv4 is $113,088$. For the base learner, $\alpha = 0.01$, $\mathbf{v}^{(0)} = \frac{1}{m}\mathbf{1}$, and T_{in} is set to 5 (resp. 15) at meta-training (resp. meta-validation or meta-testing). We train the subspace bases for $T = 100,000$ 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. The temperature is set to $\gamma_t = \max(10^{-5}, 0.8 - t/T)$, which is again a linear annealing schedule (Chen et al., 2020; Zhou et al., 2021b). To prevent overfitting, we evaluate the meta-validation performance every $2,000$ iterations and stop training when the meta-validation accuracy has no significant improvement for 5 consecutive evaluations. Hyperparameters K and m are chosen from 1 to 5 on the meta-validation set. In practice, as shown in Section 4.3.3, m can simply be fixed at 2, and K can be chosen from 3 to 4. As the search space of K is small, the additional cost of tuning K and m is small.

A.3. Results on *Meta-Dataset-BTAF*

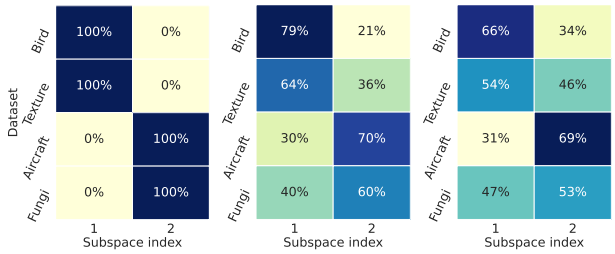
Figure 9 shows the assignment of tasks to the learned subspaces in the 5-way 1-shot setting on *Meta-Dataset-BTAF*.

B. Proofs

For notation simplicity, throughout this section, we omit the superscript τ'. Let $\mathbf{z} \equiv (\mathbf{x}, y)$ be the samples, $\ell(\mathbf{z}; \mathbf{w}) \equiv \ell(f(\mathbf{x}; \mathbf{w}), y)$, and $\nabla_{\mathbf{w}} \ell(f(\mathbf{z}; \mathbf{S}\mathbf{v})) \equiv \nabla_{\mathbf{w}} \ell(f(\mathbf{z}; \mathbf{w}))\,|_{\mathbf{w}=\mathbf{S}\mathbf{v}}$.

We first show the stability constant (in Lemma 9 of (Bousquet & Elisseeff, 2002)) in Algorithm 1 is of the order $\mathcal{O}(1/N_{tr})$.

Let $\{\mathbf{z}'_i : i = 1, \ldots, N_{tr}\}$ be another N_{tr} samples from τ. Let $\mathcal{D}_\tau^{tr(i)}$ be another training set which differs from $\mathcal{D}_\tau^{tr}$ only in the ith sample (i.e., $\mathcal{D}_\tau^{tr(i)} \equiv (\mathcal{D}_\tau^{tr} - \{\mathbf{z}_i\}) \cup \{\mathbf{z}'_i\}$). We let $\mathbf{v}_{\tau,k,i}$ (resp. $\mathbf{v}_{\tau,k}$) be the task model obtained from the base



(a) Meta-training. (b) Meta-validation. (c) Meta-testing.

Figure 9. Task assignment to the learned subspaces in 5-way 1-shot on *Meta-Dataset-BTAF* (K selected by meta-validation set is 2).

learner when using training set $\mathcal{D}_\tau^{tr(i)}$ (resp. $\mathcal{D}_\tau^{tr}$).

Lemma B.1 (Lemma 9 of (Bousquet & Elisseeff, 2002)). *Let $\mathcal{D} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ be a dataset containing m samples. For any learning algorithm $\mathcal{A}$ (receives a training set and outputs a learned model) and loss function ℓ such that $0 \leq \ell(\cdot) \leq M$, we have*

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathbf{z}}\ell(\mathbf{z};\mathcal{A}(\mathcal{D})) - \mathcal{R}(\mathcal{D};\mathcal{A}(\mathcal{D}))\right]^2 \leq \frac{M^2}{2m} + 3M\mathbb{E}_{\mathcal{D},\mathbf{z}_i'}|\ell(\mathbf{z}_i;\mathcal{A}(\mathcal{D})) - \ell(\mathbf{z}_i;\mathcal{A}(\mathcal{D}^{(i)}))| \tag{6}$$

for any $i \in \{1, \ldots, m\}$, where $\mathcal{D}^{(i)}$ is a dataset obtained by replacing $\mathbf{z}_i$ with $\mathbf{z}_i'$, and $\mathcal{R}(\mathcal{D};\mathcal{A}(\mathcal{D}))$ is the empirical risk.

Lemma B.2. *For the base learner in Algorithm 1, we have $\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathbb{E}_{\mathbf{z}_i'\sim\tau}|\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}_{\tau,k}),y_i) - \ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}_{\tau,k,i}),y_i)| \leq \frac{2\varrho(1+\alpha\beta\rho^2 m)^{T_{in}}}{N_{tr}}$.*

Proof. **Claim 1**: For $k \in \{1, \ldots, K\}$ and $i \in \{1, \ldots, N_{tr}\}$, it holds that $\|\mathbf{v}_{\tau,k} - \mathbf{v}_{\tau,k,i}\| \leq \frac{2\varrho(1+\alpha\varrho\rho^2 m)^{T_{in}}}{\rho\beta\sqrt{m}N_{tr}}$.

By the update rule in the base learner, we have $\|\mathbf{v}_{\tau,k}^{(t'+1)} - \mathbf{v}_{\tau,k,i}^{(t'+1)}\| = \|\mathbf{v}_{\tau,k}^{(t')} - \alpha\nabla_{\mathbf{v}_{\tau,k}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')}) - \mathbf{v}_{\tau,k,i}^{(t')} + \alpha\nabla_{\mathbf{v}_{\tau,k,i}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr(i)};\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')})\| \leq \|\mathbf{v}_{\tau,k}^{(t')} - \mathbf{v}_{\tau,k,i}^{(t')}\| + \alpha\|\nabla_{\mathbf{v}_{\tau,k}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')}) - \nabla_{\mathbf{v}_{\tau,k,i}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr(i)};\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')})\|$. For the second term, by the chain rule, it follows that

$$\begin{aligned}
&\left\|\nabla_{\mathbf{v}_{\tau,k}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')}) - \nabla_{\mathbf{v}_{\tau,k,i}^{(t')}}\mathcal{L}(\mathcal{D}_\tau^{tr(i)};\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')})\right\| \\
=&\left\|\mathbf{S}_k^\top\left(\nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')}) - \nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_\tau^{tr(i)};\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')})\right)\right\| \\
\leq&\|\mathbf{S}_k\|_{\mathrm{F}}\cdot\|\frac{1}{N_{tr}}\sum_{j\neq i}\left(\nabla_{\mathbf{w}}\ell(f(\mathbf{z}_j;\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')})) - \nabla_{\mathbf{w}}\ell(f(\mathbf{z}_j;\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')}))\right) \\
&+\frac{1}{N_{tr}}\left(\nabla_{\mathbf{w}}\ell(f(\mathbf{z}_i;\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')})) - \nabla_{\mathbf{w}}\ell(f(\mathbf{z}_i';\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')}))\right)\| &(7)\\
\leq&\rho\sqrt{m}\Bigg(\frac{1}{N_{tr}}\sum_{j\neq i}\left\|\nabla_{\mathbf{w}}\ell(f(\mathbf{z}_j;\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')})) - \nabla_{\mathbf{w}}\ell(f(\mathbf{z}_j;\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')}))\right\| \\
&+\frac{1}{N_{tr}}\left\|\nabla_{\mathbf{w}}\ell(f(\mathbf{z}_j;\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')}))\right\| + \frac{1}{N_{tr}}\left\|\nabla_{\mathbf{w}}\ell(f(\mathbf{z}_i';\mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')}))\right\|\Bigg) &(8)\\
\leq&\rho\sqrt{m}\left(\frac{1}{N_{tr}}\sum_{j\neq i}\beta\|\mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')} - \mathbf{S}_k\mathbf{v}_{\tau,k,i}^{(t')}\| + \frac{2\varrho}{N_{tr}}\right) &(9)\\
\leq&\rho\sqrt{m}\left(\frac{N_{tr}-1}{N_{tr}}\beta\|\mathbf{S}_k\|\|\mathbf{v}_{\tau,k}^{(t')} - \mathbf{v}_{\tau,k,i}^{(t')}\| + \frac{2\varrho}{N_{tr}}\right) &(10)\\
\leq&m\rho^2\beta\|\mathbf{v}_{\tau,k}^{(t')} - \mathbf{v}_{\tau,k,i}^{(t')}\| + \frac{2\varrho\rho\sqrt{m}}{N_{tr}}, &(11)
\end{aligned}$$

where Eq.(7) uses the norm inequality $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$ and $\|\mathbf{S}_k\| \leq \|\mathbf{S}_k\|_{\mathrm{F}}$, Eq.(8) uses the compactness assumption (thus $\|\mathbf{S}_k\|_{\mathrm{F}} \leq \rho\sqrt{m}$) and the triangle inequality, Eq.(9) uses the Lipschitzness of $\nabla_{\mathbf{w}}\ell(f(\mathbf{x};\mathbf{w}),y)$, Eq.(10) uses the

Lipschitzness of $\ell(f(\mathbf{x};\mathbf{w}),y)$, and Eq.(11) uses the boundedness of $\|\mathbf{S}_k\|$ again. Hence, we obtain a recursive inequality

$$\|\mathbf{v}_{\tau,k}^{(t'+1)}-\mathbf{v}_{\tau,k,i}^{(t'+1)}\|\leq(1+\alpha m\rho^2\beta)\|\mathbf{v}_{\tau,k}^{(t')}-\mathbf{v}_{\tau,k,i}^{(t')}\|+\frac{2\alpha\varrho\rho\sqrt{m}}{N_{tr}}. \tag{12}$$

By induction, we obtain a bound for $\mathbf{v}_{\tau,k}^{(T_{in})}-\mathbf{v}_{\tau,k,i}^{(T_{in})}$:

$$\|\mathbf{v}_{\tau,k}^{(T_{in})}-\mathbf{v}_{\tau,k,i}^{(T_{in})}\|\leq(1+\alpha m\rho^2\beta)\|\mathbf{v}_{\tau,k}^{(0)}-\mathbf{v}_{\tau,k,i}^{(0)}\|+\frac{2\alpha\varrho\rho\sqrt{m}}{N_{tr}}\sum_{t'=0}^{T_{in}-1}(1+\alpha\beta\rho^2m)^{t'}\leq\frac{2\varrho(1+\alpha\varrho\rho^2m)^{T_{in}}}{\rho\beta\sqrt{m}N_{tr}}, \tag{13}$$

where we have used the fact $\mathbf{v}_{\tau,k}^{(0)}=\mathbf{v}_{\tau,k,i}^{(0)}$.

Claim 2: The stability constant of the base learner is $\frac{2\varrho(1+\alpha\beta\rho^2m)^{T_{in}}}{N_{tr}}$.

Next, we analyze the stability constant of the base learner:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathbb{E}_{\mathbf{z}_i'\sim\tau}\left|\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}_{\tau,k}),y_i)-\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}_{\tau,k,i}),y_i)\right|\\
\leq&\beta\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathbb{E}_{\mathbf{z}_i'\sim\tau}\|\mathbf{S}_k\mathbf{v}_{\tau,k,i}-\mathbf{S}_k\mathbf{v}_{\tau,k}\| &(14)\\
\leq&\beta\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathbb{E}_{\mathbf{z}_i'\sim\tau}\|\mathbf{S}_k\|_{\mathrm{F}}\|\mathbf{v}_{\tau,k,i}-\mathbf{v}_{\tau,k}\| &(15)\\
\leq&\beta\rho\sqrt{m}\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathbb{E}_{\mathbf{z}_i'\sim\tau}\|\mathbf{v}_{\tau,k,i}-\mathbf{v}_{\tau,k}\| &(16)\\
\leq&\beta\rho\sqrt{m}\cdot\frac{2\varrho(1+\alpha\beta\rho^2m)^{T_{in}}}{\rho\beta\sqrt{m}N_{tr}} &(17)\\
=&\frac{2\varrho(1+\alpha\beta\rho^2m)^{T_{in}}}{N_{tr}}, &(18)
\end{aligned}$$

where Eq.(14) uses the Lipschitz property of ℓ, Eq.(15) uses the norm inequality, Eq.(16) uses the boundedness of $\|\mathbf{S}_k\|_{\mathrm{F}}$, Eq.(17) uses the inequality (13). The above equality reveals that the stability constant in Theorem 11 of (Bousquet & Elisseeff, 2002) (β_2 there) is $\frac{2\varrho(1+\alpha\beta\rho^2m)^{T_{in}}}{N_{tr}}$.

□

B.1. Proof of Theorem 3.3

Proof. The proof is based on the connection between generalization and stability (Bousquet & Elisseeff, 2002).

We adopt the notations used in the proof of Lemma B.2. We apply Lemma B.1 to our algorithm and obtain

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x};\mathbf{S}_k\mathbf{v}_{\tau,k}),y)-\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k})\right]^2\\
\leq&\frac{\nu^2}{2N_{tr}}+3\nu\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathbb{E}_{\mathbf{z}_i'\sim\tau}\left|\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}_{\tau,k}),y_i)-\ell(f(\mathbf{x}_i;\mathbf{S}_k\mathbf{v}_{\tau,k,i}),y_i)\right|\\
\leq&\frac{\nu^2}{2N_{tr}}+\frac{6\nu\varrho(1+\alpha\beta\rho^2m)^{T_{in}}}{N_{tr}}, &(19)
\end{aligned}$$

where (19) uses the equality (18) in Lemma B.2. By the Cauchy-Schwarz inequality, we have

$$\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left|\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x};\mathbf{S}_k\mathbf{v}_{\tau,k}),y)-\mathcal{L}(\mathcal{D}_\tau^{tr};\mathbf{S}_k\mathbf{v}_{\tau,k})\right|\leq\sqrt{\frac{\nu^2}{2N_{tr}}+\frac{6\nu\varrho(1+\alpha\beta\rho^2m)^{T_{in}}}{N_{tr}}}. \tag{20}$$

To provide an upper bound of $\mathcal{R}(\mathcal{S}) - \hat{\mathcal{R}}(\mathcal{S})$, we need to address the randomness in k_τ:

$$\begin{aligned}
\mathcal{R}(\mathcal{S}) - \hat{\mathcal{R}}(\mathcal{S}) =& \mathbb{E}_\tau \mathbb{E}_{\mathcal{D}_\tau^{tr}} \left[\mathbb{E}_{\mathbf{z}\sim\tau} \ell(f(\mathbf{x}; \mathbf{S}_{k_\tau} \mathbf{v}_{\tau,k_\tau}), y) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau} \mathbf{v}_{\tau,k_\tau}) \right] \\
=& \mathbb{E}_\tau \mathbb{E}_{\mathcal{D}_\tau^{tr}} \sum_{k=1}^{K} \mathbb{I}_{[k_\tau=k]} \left[\mathbb{E}_{\mathbf{z}\sim\tau} \ell(f(\mathbf{x}; \mathbf{S}_k \mathbf{v}_{\tau,k}), y) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}) \right] \\
\leq& \mathbb{E}_\tau \sum_{k=1}^{K} \mathbb{E}_{\mathcal{D}_\tau^{tr}} \mathbb{I}_{[k_\tau=k]} \left| \mathbb{E}_{\mathbf{z}\sim\tau} \ell(f(\mathbf{x}; \mathbf{S}_k \mathbf{v}_{\tau,k}), y) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}) \right| \\
\leq& \mathbb{E}_\tau \sum_{k=1}^{K} \mathbb{E}_{\mathcal{D}_\tau^{tr}} \left| \mathbb{E}_{\mathbf{z}\sim\tau} \ell(f(\mathbf{x}; \mathbf{S}_k \mathbf{v}_{\tau,k}), y) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}) \right| \\
\leq& K \sqrt{\frac{\nu^2}{2N_{tr}} + \frac{6\nu\varrho(1+\alpha\beta\rho^2 m)^{T_{in}}}{N_{tr}}},
\end{aligned}$$

where the first inequality is because the empirical loss can be smaller than the population loss, and the last inequality follows from the Eq.(20). □

B.2. Proof of Theorem 3.5

Proof of Theorem 3.5. We first show that $\mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v})$ satisfies the PL inequality in $\mathbf{v}$. By the chain rule, it follows that

$$\begin{aligned}
\|\nabla_{\mathbf{v}} \mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v})\|^2 = \|\mathbf{S}^\top \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v})\|^2 &= \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v})^\top \mathbf{S}\mathbf{S}^\top \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v}) \\
&\geq \lambda_{\min}\left(\mathbf{S}\mathbf{S}^\top\right) \|\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v})\|^2 \\
&\geq c_1^2 \left(1 - c_2\sqrt{\frac{m}{d}}\right)^2 \|\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v})\|^2 \qquad (21) \\
&\geq \lambda c_1^2 \left(1 - c_2\sqrt{\frac{m}{d}}\right)^2 (\mathcal{L}(\mathcal{D}; \mathbf{S}_k \mathbf{v}) - L_\star), \qquad (22)
\end{aligned}$$

where Eq.(21) uses Assumption 3.4(ii), and Eq.(22) uses Assumption 3.4(i).

Then, we show that $\nabla_{\mathbf{v}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v})$ is $(\beta\rho^2 m)$-Lipschitz in $\mathbf{v}$. Using the chain rule, we have

$$\begin{aligned}
\|\nabla_{\mathbf{v}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}) - \nabla_{\mathbf{v}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}')\| &= \|\mathbf{S}_k^\top \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v})| - \mathbf{S}_k^\top \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}')\| \\
&\leq \|\mathbf{S}_k\| \|\nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v})| - \nabla_{\mathbf{w}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}')\| \qquad (23) \\
&\leq \beta \|\mathbf{S}_k\| \|\mathbf{S}_k \mathbf{v} - \mathbf{S}_k \mathbf{v}'\| \qquad (24) \\
&\leq \beta \|\mathbf{S}_k\|^2 \|\mathbf{v} - \mathbf{v}'\| \qquad (25) \\
&\leq \beta\rho^2 m \|\mathbf{v} - \mathbf{v}'\|, \qquad (26)
\end{aligned}$$

where Eqs.(23) and (24) use the norm inequality $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$, Eq.(24) uses the Assumption 3.2(i) on smoothness, Eq.(24) uses $\|\mathbf{S}_k\|^2 \leq \|\mathbf{S}_k\|_F^2 \leq \rho^2 m$ by Assumption 3.2(ii) on compactness.

Using the Taylor expansion, we obtain

$$\begin{aligned}
\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t'+1)}) &= \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t')}) + (\mathbf{v}_{\tau,k}^{(t'+1)} - \mathbf{v}_{\tau,k}^{(t')})^\top \nabla_{\mathbf{v}_{\tau,k}^{(t')}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t')}) \\
&\quad + \frac{1}{2} (\mathbf{v}_{\tau,k}^{(t'+1)} - \mathbf{v}_{\tau,k}^{(t')})^\top \nabla^2_{\mathbf{v}_{\tau,k}^{(t')}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \boldsymbol{\xi}^{(t')}) (\mathbf{v}_{\tau,k}^{(t'+1)} - \mathbf{v}_{\tau,k}^{(t')}) \\
&\leq \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t')}) - \alpha \left\| \nabla_{\mathbf{v}_{\tau,k}^{(t')}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t')}) \right\|^2 + \frac{1}{2} \alpha^2 \beta \rho^2 m \left\| \nabla_{\mathbf{v}_{\tau,k}^{(t')}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t')}) \right\|^2 \qquad (27) \\
&\leq \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t')}) - \frac{\alpha}{2} \left\| \nabla_{\mathbf{v}_{\tau,k}^{(t')}} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k \mathbf{v}_{\tau,k}^{(t')}) \right\|^2, \qquad (28)
\end{aligned}$$

where $\boldsymbol{\xi}_{t'} \in [\mathbf{v}_{\tau,k}^{(t')}, \mathbf{v}_{\tau,k}^{(t'+1)}]$, Eq.(27) uses $\|\nabla_{\mathbf{v}}^2 \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v})\|^2 \leq \beta\rho^2 m$ follows from Claim 1, Eq.(28) uses $\alpha < \frac{1}{\beta\rho^2 m}$ by assumption. By Assumption 3.4, it follows that

$$\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}^{(t'+1)}) - L_\star \leq \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')}) - L_\star - \frac{\alpha}{2}\lambda c_1^2 \left(1 - c_2\sqrt{\frac{m}{d}}\right)^2 (\mathcal{L}(\mathcal{D}; \mathbf{S}_k\mathbf{v}) - L_\star) \tag{29}$$

$$\leq (1-\psi)\left(\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}^{(t')}) - L_\star\right), \tag{30}$$

where $\psi = \frac{\alpha\lambda c_1^2}{2}\left(1 - c_2\sqrt{\frac{m}{d}}\right)^2$.

The above recursive inequality implies $\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}) - L_\star = \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}^{(T_{in})}) - L_\star \leq (1-\psi)^{T_{in}}\left(\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}^{(0)}) - L_\star\right)$. As $\ell(\cdot,\cdot)$ is bounded by ν, $\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}^{(0)}) = \frac{1}{|\mathcal{D}_\tau^{tr}|}\sum_{(\mathbf{x},y)\in\mathcal{D}_\tau^{tr}} \ell(f(\mathbf{x}; \mathbf{S}_k\mathbf{v}^{(0)}), y)$ is also bounded by ν, then we have $\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}) \leq L_\star + \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k}^{(T_{in})}) \leq L_\star + (1-\psi)^{T_{in}}\left(\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}^{(0)}) - L_\star\right) \leq L_\star + (1-\psi)^{T_{in}}(\nu - L_\star) \leq L_\star + (1-\psi)^{T_{in}}\nu$. Let $L_k \equiv \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_k\mathbf{v}_{\tau,k})$. Next, we consider two events $\mathcal{E} \equiv \{\min_{1\leq k\leq K} L_k \geq L_\star + \frac{\nu}{K+\nu}\}$ and $\bar{\mathcal{E}} \equiv \{\min_{1\leq k\leq K} L_k < L_\star + \frac{\nu}{K+\nu}\}$. We have $\mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}} \min_{1\leq k\leq K} L_k = \mathbb{E}_\tau\mathbb{E}_{\mathcal{E}} \min_{1\leq k\leq K} L_k + \mathbb{E}_\tau\mathbb{E}_{\bar{\mathcal{E}}} \min_{1\leq k\leq K} L_k \leq \mathbb{E}_\tau\left(L_\star + (1-\psi)^{T_{in}}\nu\right)\mathbb{P}(\mathcal{E}) + \mathbb{E}_\tau(L_\star + \frac{\nu}{K+\nu})\mathbb{P}(\bar{\mathcal{E}}) \leq 2L_\star + (1-\psi)^{T_{in}}\nu + \frac{\nu}{K+\nu} \leq \sqrt{4L_\star^2 + 2\nu^2(1-\psi)^{2T_{in}}} + \frac{\nu}{K+\nu}$, where we have used the property $\mathbb{P}(\bar{\mathcal{E}}) \leq 1$. As $\psi = \frac{\alpha\lambda c_1^2}{2}\left(1 - c_2\sqrt{\frac{m}{d}}\right)^2$, it follows that $(1-\psi)^{2T_{in}} = \left(c_1c_2\sqrt{\alpha\lambda/(2d)}\left(\sqrt{m} - (\sqrt{d}/c_2)(\sqrt{2/(c_1^2\alpha\lambda)}+1)\right)\left(\sqrt{m} + (\sqrt{d}/c_2)(\sqrt{2/(c_1^2\alpha\lambda)}-1)\right)\right)^{2T_{in}} = ((\sqrt{m} - \kappa_1)(\sqrt{m} + \kappa_2)\phi)^{2T_{in}}$, where $\kappa_1 = (\sqrt{d}/c_2)(\sqrt{2/(c_1^2\alpha\lambda)} + 1)$, $\kappa_2 = (\sqrt{d}/c_2)(\sqrt{2/(c_1^2\alpha\lambda)} - 1)$, and $\phi = c_1c_2\sqrt{\alpha\lambda/(2d)}$. Note that κ_1, κ_2, and ϕ are all positive. Hence, we obtain $\mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}} \min_{1\leq k\leq K} L_k \leq \sqrt{4L_\star^2 + 2\nu^2((\sqrt{m} - \kappa_1)(\sqrt{m} + \kappa_2)\phi)^{2T_{in}}} + \frac{\nu}{K+\nu}$ and finish the proof. □

B.3. Proof of Theorem 3.7

Proof. By the definition of excess risk, we have

$$\begin{aligned}
0 &\leq \mathcal{R}(\mathcal{S}) - \mathcal{R}^\star \\
&= \mathbb{E}_\tau[\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}), y) - \mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}) + \mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}) - \mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star) \\
&\quad + \mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star) - \mathbb{E}_{\mathbf{z}\sim\tau}\ell(\mathbf{z}; \mathbf{w}_\tau^\star)] \qquad (31) \\
&= \mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}), y) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau})\right] + \mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star)\right] \\
&\quad + \mathbb{E}_\tau\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,k_\tau}^\star), y) - \mathbb{E}_{\mathbf{z}\sim\tau}\ell(f(\mathbf{x}; \mathbf{w}_\tau^\star), y)\right] \\
&\leq K\sqrt{\frac{\nu^2}{2N_{tr}} + \frac{6\nu\varrho(1+\alpha\beta\rho^2 m)^{T_{in}}}{N_{tr}}} + \mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star)\right] \\
&\quad + \mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{w}_{\tau,\,\mathbf{S}_{k_\tau}}^\star)\right] + \mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\|\nabla_{\mathbf{w}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \boldsymbol{\xi}_\tau)\|\|\mathbf{w}_{\tau,\mathbf{S}_{k_\tau}^\perp}^\star\| \qquad (32) \\
&\leq K\sqrt{\frac{\nu^2}{2N_{tr}} + \frac{6\nu\varrho(1+\alpha\beta\rho^2 m)^{T_{in}}}{N_{tr}}} + \mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star)\right] + \varrho\mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\operatorname{dist}(\mathbf{w}_\tau^\star, \mathbb{S}_{k_\tau}), \\
&\qquad (33)
\end{aligned}$$

where (31) follows by introducing two additional terms ($\mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau})$ and $\mathbb{E}_\tau\mathbb{E}_{\mathcal{D}_\tau^{tr}}\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star)$), Eq.(32) uses the bound in Theorem 3.3 and the mean value theorem (we decompose $\mathbf{w}_\tau^\star = \mathbf{w}_{\tau,\mathbf{S}_{k_\tau}}^\star + \mathbf{w}_{\tau,\mathbf{S}_{k_\tau}^\perp}^\star$ and $\boldsymbol{\xi}_\tau \in [\mathbf{w}_{\tau,\mathbf{S}_{k_\tau}}^\star, \mathbf{w}_\tau^\star]$), and Eq.(33) follows from the Lipschitzness assumption and $\mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{w}_{\tau,\,\mathbf{S}_{k_\tau}}^\star)\right] \leq 0$ as $\mathbf{v}_{\tau,k_\tau}^\star$ is

an exact solution of the problem $\min_{\mathbf{v}_\tau} \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_\tau)$. We conclude that

$$\begin{aligned}
&\mathcal{R}(\mathcal{S}) - \mathcal{R}^\star \\
&\leq K\sqrt{\frac{\nu^2}{2N_{tr}} + \frac{6\nu\varrho(1+\alpha\beta\rho^2 m)^{T_{in}}}{N_{tr}}} + \mathbb{E}_\tau \mathbb{E}_{\mathcal{D}_\tau^{tr}}\left[\mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}) - \mathcal{L}(\mathcal{D}_\tau^{tr}; \mathbf{S}_{k_\tau}\mathbf{v}_{\tau,k_\tau}^\star)\right] + \varrho \mathbb{E}_\tau \mathbb{E}_{\mathcal{D}_\tau^{tr}} \operatorname{dist}(\mathbf{w}_\tau^\star, \mathbb{S}_{k_\tau}) \\
&\leq K\sqrt{\frac{\nu^2}{2N_{tr}} + \frac{6\nu\varrho(1+\alpha\beta\rho^2 m)^{T_{in}}}{N_{tr}}} + \rho\sqrt{m}\, \mathbb{E}_\tau \mathbb{E}_{\mathcal{D}_\tau^{tr}} \|\mathbf{v}_{\tau,k_\tau} - \mathbf{v}_{\tau,k_\tau}^\star\| + \varrho \mathbb{E}_\tau \mathbb{E}_{\mathcal{D}_\tau^{tr}} \operatorname{dist}(\mathbf{w}_\tau^\star, \mathbb{S}_{k_\tau}),
\end{aligned}$$

where the last inequality follows from the Lipschitzness of ℓ and $\|\mathbf{S}_{k_\tau}\| \leq \|\mathbf{S}_{k_\tau}\|_F \leq \rho\sqrt{m}$. □

B.4. Proof of Proposition 3.1

Proof. This proposition is a property of linear regression tasks and has been mentioned in (Kong et al., 2020; Tripuraneni et al., 2021). We include the proof here for completeness.

By the definition $y = \mathbf{x}^\top \mathbf{w} + \xi$, we have

$$\begin{aligned}
&\mathbb{E}_{\tau\sim p(\tau)} \mathbb{E}_{(\mathbf{x},y)\sim\tau,(\mathbf{x}',y')\sim\tau} y y' \mathbf{x}\mathbf{x}'^\top \\
=&\mathbb{E}_{\tau\sim p(\tau)} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\mathbf{x}'\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\xi\sim\mathcal{N}(0,\sigma_\xi^2),\xi'\sim\mathcal{N}(0,\sigma_\xi^2)} (\mathbf{x}^\top \mathbf{w}_\tau^\star + \xi)(\mathbf{x}'^\top \mathbf{w}_\tau^\star + \xi')\mathbf{x}\mathbf{x}'^\top \\
=&\mathbb{E}_{\tau\sim p(\tau)} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\mathbf{x}'\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\xi\sim\mathcal{N}(0,\sigma_\xi^2),\xi'\sim\mathcal{N}(0,\sigma_\xi^2)} (\mathbf{x}^\top \mathbf{w}_\tau^\star \mathbf{x}'^\top \mathbf{w}_\tau^\star \mathbf{x}\mathbf{x}'^\top + \mathbf{x}^\top \mathbf{w}_\tau^\star \xi' \mathbf{x}\mathbf{x}'^\top + \xi \mathbf{x}'^\top \mathbf{w}_\tau^\star \mathbf{x}\mathbf{x}'^\top + \xi\xi' \mathbf{x}\mathbf{x}'^\top) \\
=&\mathbb{E}_{\tau\sim p(\tau)} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\mathbf{x}'\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \mathbf{x}^\top \mathbf{w}_\tau^\star \mathbf{x}'^\top \mathbf{w}_\tau^\star \mathbf{x}\mathbf{x}'^\top,
\end{aligned}$$

where the last equality follows from the independence of $\xi, \xi', \mathbf{x}$, and $\mathbf{x}'$. Using the independence of $\mathbf{x}$, and $\mathbf{x}'$, we obtain $\mathbb{E}_{\tau\sim p(\tau)} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\mathbf{x}'\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \mathbf{x}^\top \mathbf{w}_\tau^\star \mathbf{x}'^\top \mathbf{w}_\tau^\star \mathbf{x}\mathbf{x}'^\top = \mathbb{E}_{\tau\sim p(\tau)} (\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \mathbf{x}\mathbf{x}^\top) \mathbf{w}_\tau^\star \mathbf{w}_\tau^{\star\top} (\mathbb{E}_{\mathbf{x}'\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \mathbf{x}'\mathbf{x}'^\top) = \mathbb{E}_{\tau\sim p(\tau)} \mathbf{w}_\tau^\star \mathbf{w}_\tau^{\star\top}$. □