Pre-Trained Model Reusability Evaluation for Small-Data Transfer Learning

Yao-Xiang Ding
State Key Lab for CAD&CG
Zhejiang University
`yxding@zju.edu.cn`

Xi-Zhu Wu
National Key Lab for Novel Software Technology
Nanjing University
`wuxz@lamda.nju.edu.cn`

Kun Zhou
State Key Lab for CAD&CG
Zhejiang University
`kunzhou@acm.org`

Zhi-Hua Zhou
National Key Lab for Novel Software Technology
Nanjing University
`zhouzh@nju.edu.cn`

Abstract

We study *model reusability evaluation* (MRE) for source pre-trained models: evaluating their transfer learning performance to new target tasks. In special, we focus on the setting under which the target training datasets are small, making it difficult to produce reliable MRE scores using them. Under this situation, we propose *synergistic learning* for building the task-model metric, which can be realized by collecting a set of pre-trained models and asking a group of data providers to participate. We provide theoretical guarantees to show that the learned task-model metric distances can serve as trustworthy MRE scores, and propose synergistic learning algorithms and models for general learning tasks. Experiments show that the MRE models learned by synergistic learning can generate significantly more reliable MRE scores than existing approaches for small-data transfer learning.

1 Introduction

Reusing pre-trained models have played essential roles in modern learning pipelines for decreasing training cost, alleviating the requirement of big datasets, and reducing the danger of catastrophic forgetting. The growing number of pre-trained models promotes the birth of large pre-trained model zoos, making it closer towards the future learnware market [Zhou and Tan, 2022]. When selecting a model from these model zoos for doing model transfer, one has to do model reusability evaluation (MRE) first: evaluating the transfer learning performance of the models to the target task and identifying the best model. The role of MRE is crucial since no matter how good the transfer learning strategy is, incorrect MRE would still lead to the danger of negative transfer. MRE has received growing attention in recent years [Achille et al., 2019, Tran et al., 2019, Nguyen et al., 2020, Wu et al., 2020, Ding and Zhou, 2020, You et al., 2021]. But most existing studies focus on large-data MRE, under which the target dataset is sufficiently large.

In this work, we focus on small-data MRE, under which the target training datasets are small. Reusing pre-trained models is essential under the small-data scenario since learning from scratch is difficult. Unfortunately, large-data MRE approaches are usually invalid for small data since they usually focus more on simplicity and efficiency, but not generalization and robustness, which are essential for small-data MRE. It is indeed challenging to obtain reliable MRE results under the small-data scenario due to the fundamental burden set by the laws of statistics.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

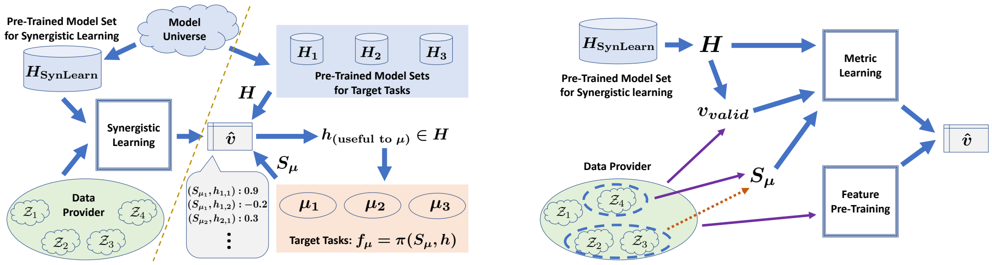


Figure 1: The illustration of the proposed approach. The left figure illustrates the overall procedure, in which the left part illustrates the synergistic learning stage and the right part illustrate the testing MRE stage. The right figure illustrates Algorithm 1. The arrows started from the data providers denote the queries, in which the solid purple ones are `SQUERY` and the dashed orange one is `MQUERY`.

Our solution is based on two observations. First, model reusability usually depends on the task-model relationship other than the task-independent property of any specific small sample. We could grasp this relationship by meta-learning an MRE function among tasks and models. Second, this learning process is often realizable in practice, in special for large model zoo platforms where sufficient pre-trained models and data providers are available. Based on these observations, we conduct the following studies in this paper:

Problem formuation. We provide the formulation of small-data MRE and synergistic learning to learn the MRE function by metric learning. Synergisitc learning works for general learning scenarios beyond classification, on which most previous MRE approaches focus.

Theoretical analysis. We propose access risk analysis showing that the MRE model learned by synergistic learning guarantees to generate reliable MRE scores. The theory not only provides guarantee even for using non-convex deep networks as the MRE model, but also motivates feature pre-training in synergistic learning.

Algorithm and model design. We propose synergistic learning algorithms and MRE model structures for general learning tasks, which have the auxiliary advantage of protecting data privacy. Meanwhile, we propose a more elaborate MRE model strucuture for classification.

Experimental verification. Experimental results show that synergistic learning can generate significantly more reliable MRE scores for small-data transfer learning than existing MRE approaches.

2 Problem Setup

In this section, we provide the formal definition of small-data MRE and synergistic learning.

2.1 Small-Data MRE

Denote by $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ the observation space, in which $\mathcal{X}, \mathcal{Y}$ are the input and output spaces respectively. A target learning task μ can be represented by a probability measure[1] $\mu(x, y)$ over some $\mathcal{Z}_\mu \subseteq \mathcal{Z}$. Furthermore, we assume that all μ are drawn from the task environment $\mathcal{T}$, a probability measure over the space of all target tasks.

In small-data MRE, target tasks μ are drawn from $\mathcal{T}$. For each μ, the objective is to learn a prediction rule f_μ such that for any observation $(x, y) \sim \mu$, $f_\mu(x)$ is close to y. The learner of μ will be given a target training dataset S_μ, a set of pre-trained models H, a model transfer strategy π and an MRE function v_π. $S_\mu = \{z_k = (x_k, y_k)\}_{k=1}^K$ is a sample drawn from μ such that the sample size K is a small number. H is the pre-trained model set in which each model $h \in H$ is drawn from the model environment $\mathcal{M}$, a probability measure over the space of all pre-trained models. We assume that the models in H are sampled independently from μ while different H can be given for different μ. $\pi(S_\mu, h)$ is a transfer learning strategy which can be used to transfer a pre-trained model h into f_μ using S_μ. We do not restrict the choice of π, but assume that π is fixed for any target task. The MRE function $v_\pi(S_\mu, h)$ is a real-valued function taking S_μ and h as the inputs

---

[1] We will use μ to denote both a task and its probability measure below when there is no ambiguity.

and outputing the transferability score. Below we use v instead of v_π to simplify notations. Without loss of generality, among all possible v, we define v^* as the optimal MRE function such that for any μ, S_μ, h_1, h_2, $v^*(S_\mu, h_1) < v^*(S_\mu, h_2)$ if and only if h_1 has better transfer learning performance with S_μ than h_2. Thus we assume that the MRE score is *monotonically decreasing* w.r.t. transfer performance. If no special properties exist for v^*, estimating v^* would be extremely difficult for small-data MRE. Fortunately, in general, we could assume that model reusability usually depends on the task-model relationship other than the task-independent property of any specific small sample, as discussed in Section 1. Therefore, we assume that *the ground-truth model transferability depends only on the target task μ but does not depend on the random draw of S_μ from μ*, i.e. $\forall S_\mu, S'_\mu \sim \mu, v^*(S_\mu) = v^*(S'_\mu)$. Thus we can use the notation $v^*(\mu, h)$ instead of $v^*(S_\mu, h)$ for any S_μ, h. How this assumption can be relaxed is discussed in Section 7.

2.2 Synergistic Learning

Synergistic learning is the preparation stage for learning an MRE function $\hat{v}$ that accurately approximate v^* before any MRE problem defined in Section 2.1 starts. A synergistic learning process works with the following three prerequisites:

- A set of pre-trained models $H_{\text{SynLearn}} = \{h_k\}_{k=1}^{N_m}$ drawn from $\mathcal{M}$ are given. Note that the models in this set could be different from models given in any future MRE problems;
- A set of data providers $\mathcal{D} = \{\mathcal{Z}_d\}_{d=1}^{D}$ participate in learning, such that each data provider represents an observation subspace $\mathcal{Z}_d \subseteq \mathcal{Z}$. Furthermore, $\cup \mathcal{Z}_d = \mathcal{Z}$. We define the closure of any task $\mu \sim \mathcal{T}$ as $\bar{\mathcal{Z}}_\mu = \min_L \{\mathcal{Z}_{d_1}, \mathcal{Z}_{d_2}, \dots \mathcal{Z}_{d_L}\}$ s.t. $\mathcal{Z}_\mu \subseteq \cup_{l=1}^{L} \mathcal{Z}_{d_l}$.
- A validation MRE function v_{valid} is provided for synergistic learning. v_{valid} is unbiased between any task and model[2]: $\forall \mu, h, v_{valid}(\mu, h) = \mathbb{E}_{S_\mu \sim \mu}[v_{valid}(S_\mu, h)] = v^*(\mu, h)$.

The basic idea of synergistic learning is to establish a metric $\hat{v}(S_\mu, h)$ between the target training datasets and the models so that the metric distance could be used as the MRE score. In Section 4, we will show that learning $\hat{v}$ only requires the data providers to answer two kinds of queries:

- `SQUERY`$(\psi_s, \mathcal{Z}_d, \mu, M)$: a *single* data provider $\mathcal{Z}_d$ is queried to sample M observations from a given probability measure μ over $\mathcal{Z}_d$, and return the output of $\psi_s(\{z_i\}_{i=1}^{M})$, a function of the observations.
- `MQUERY`$(\psi_m, \{\mathcal{Z}_{d_i}\}_{i=1}^{Q}, \{\mu_i\}_{i=1}^{Q}, \{M_i\}_{i=1}^{Q})$: *multiple* data providers $\{\mathcal{Z}_{d_i}\}_{i=1}^{Q}$ are queried jointly. Each data provider $\mathcal{Z}_{d_i}$ sample M_i observations from a given probability measure μ_i over $\mathcal{Z}_{d_i}$. Aggregating the sampled observations from all data providers, the output of $\psi_m(\{z_{1,j}\}_{j=1}^{M_1}, \{z_{2,j}\}_{j=1}^{M_2}, \dots, \{z_{Q,j}\}_{j=1}^{M_Q})$, a function of all observations, is returned. Furthermore, ψ_m cannot be realized by aggregating multiple `SQUERY`.

The main difference between these two queries lies in the scope of involved data providers, which is important when the data privacy is sensitive. `SQUERY` is answered by a single data provider. How this kind of queries can be answered in privacy-guaranteed ways has received many studies [Zinkevich et al., 2010, Konečný et al., 2016]. In contrast, `MQUERY` needs to be answered by multiple data providers jointly. Protecting data privacy is much harder in this situation since their data need to be aggregated. Therefore, we set an auxiliary target of reducing the number of times to use `MQUERY`.

Finally, for learning $\hat{v}$, we assume that $\hat{v}$ is formed by three parts: (1) $g_\mu(S_\mu)$, the task feature backbone; (2) $g_h(h)$, the combination of the model specification generator[3] and the model feature backbone; (3) $d_\theta(c_\mu, c_h)$, the metric module, in which c_μ is the output of $g_\mu(S_\mu)$, c_h is the output of $g_h(h)$, and θ is the learnable parameter of $d_\theta(c_\mu, c_h)$. g_μ and g_h transfer target training dataset S_μ and pre-trained model h into their representations c_μ and c_h. The metric module then calculates the metric distance between c_μ and c_h as the MRE score. The details are introduced in Section 4.2.

---

[2] Even though v_{valid} is unbiased, it may have high variance when S_μ is small. Thus v_{valid} can not be used directly for small-data MRE.

[3] Please refer to Section 7 for details about model specifications.

3 Theoretical Analysis

In this section, we discuss the theoretical foundation of synergistic learning. Readers who are interested more on algorithmic ideas can skip this section without affecting understanding significantly. In the analysis, we assume that g_μ and g_h are fixed and focus on learning the metric module. Thus the learnable parameter for $\hat{v}$ is θ.

For metric-based synergistic learning, we assume that N_m pre-trained models $\{h_k\}_{k=1}^{N_m}$ sampled from $\mathcal{M}$, N_t tasks $\{\mu_i\}_{i=1}^{N_t}$ sampled from $\mathcal{T}$ and N_S target training datasets $\{S_{\mu_i,j}\}_{j=1}^{N_S}$ sampled from each task μ_i are used for training. We also assume that for any pair of h_i and $S_{\mu_j,k}$, the ground-truth MRE score $v^*(\mu_i, h_k)$ are given. Define $r_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v})$ as $\mathbf{I}[\Delta_{v^*}(\mu,\mu',h,h') < 0]\mathbf{I}[\Delta_{\hat{v}}(S_\mu, S_{\mu'}, h, h') \geq 0]$ in which $\Delta_{v^*}(\mu,\mu',h,h') = v^*(\mu,h) - v^*(\mu',h')$, $\Delta_{\hat{v}}(S_\mu, S_{\mu'}, h, h') = \hat{v}(S_\mu, h) - \hat{v}(S_{\mu'}, h')$, and $\mathbf{I}$ is the indicator function. Since v^* is the ground-truth MRE function, a desirable $\hat{v}$ should minimize

$$R(\hat{v}) = \mathbb{E}\big[r_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v})\big],$$

where the expectation is taken over $S_\mu \sim \mu, S_{\mu'} \sim \mu', \mu, \mu' \sim \mathcal{T}, h, h' \sim \mathcal{M}$. While for the convenience of optimization and analysis, we define our objective using the triplet surrogate loss. Define $r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v})$ as $\mathbf{I}[\Delta_{v^*}(\mu,\mu',h,h') < 0][\Delta_{\hat{v}}(S_\mu, S_{\mu'}, h, h') + \gamma]_+$ in which $[x]_+ = \max\{x, 0\}$ and $\gamma > 0$. When $\gamma > 1$, $r^\gamma_{v^*}$ upper bounds r_{v^*}. Meanwhile, $r^\gamma_{v^*}$ is consistent with r_{v^*} since $r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}) = 0$ indicates that $r_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}) = 0$. Therefore, we define our goal as minimizing the following expected risk

$$R^\gamma(\hat{v}) = \mathbb{E}\big[r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v})\big]. \tag{1}$$

A possible way to achieve this goal is to minimize the following empirical risk

$$\hat{R}^\gamma(\hat{v}) = \frac{1}{(N_t N_S N_m)^2} \sum \Big[r^\gamma_{v^*}(S_\mu, S_{\mu'}, h_k, h_{k'}; \hat{v})\Big] \tag{2}$$

where the summation is taken over all models, tasks and datasets used for training. We want to know whether minimizing the empirical risk $\hat{R}^\gamma(\hat{v})$ indeed leads to the minimization of the expected risk $R^\gamma(\hat{v})$. Denote by Θ the parameter space for θ and use $\hat{v}_\theta$ to denote the $\hat{v}$ with learnable parameter θ. We provide an access risk bound showing the effectiveness of minimizing the above empirical risk. Use $\bar{G}_\theta(S_\mu, S_{\mu'}, h, h')$ to denote $[\Delta_{\hat{v}_\theta}(S_\mu, S_{\mu'}, h, h') + \gamma]_+$. Additionally, we assume that $\nabla[\bar{G}_\theta(S_\mu, S_{\mu'}, h, h')] = 0$ when $\bar{G}_\theta(S_\mu, S_{\mu'}, h, h') = 0$. Furthermore, let

$$\nabla[\hat{R}(\bar{G}_\theta)] = \frac{1}{(N_t N_S N_m)^2} \sum \|\nabla^-[S_\mu, S_{\mu'}, h, h']\|,$$

in which the summation is taken over all tasks, datasets and models used for training and we have $\nabla^-[S_\mu, S_{\mu'}, h, h'] = \mathbf{I}[\Delta_{v^*}(\mu,\mu',h,h') < 0]\nabla[\bar{G}_\theta(S_\mu, S_{\mu'}, h, h')]$. Meanwhile, we denote $V = \max_{i\in[N_t]} \mathbb{E}_{S_{\mu_i}\sim\mu_i}[c^2(S_{\mu_i})]$. Now we are ready to state the following theorem whose proof is provided in the appendix (Section A).

Theorem 3.1. *Under Assumption A.1, there exist optimal parameter $\theta^* \in \Theta$ and constant $\beta > 0$, $\forall\theta \in \Theta$, the following event*

$$R^\gamma(\hat{v}_\theta) - R^\gamma(\hat{v}_{\theta^*}) \leq \beta\big[\nabla[\hat{R}(\bar{G}_\theta)] + \Delta(N_t, N_m, N_S)\big]$$

holds with high probability, in which $\Delta(N_t, N_m, N_S) = \tilde{O}(1/\sqrt{N_t}, 1/\sqrt{N_m}, V/\sqrt{N_S}, 1/N_S)$.

Theorem 3.1 shows several interesting insights. First, $\nabla[\hat{R}(\bar{G}_\theta)]$ is the gradient norm of the metric distance gaps from the training data. The bound shows that the access risk will be small when the gradient norm tends to zero, even when the global minimum of the empirical risk has not been reached. This makes the bound informative when non-convex models, such as DNNs, are used. Second, the bound has the normal $O(1/\sqrt{N_t}), O(1/\sqrt{N_m})$ sample complexity dependence for both tasks and models. This is in agreement with our intuition such that lacking any of the tasks and models would lead to the failure of learning. Finally, the most interesting take-away is the sample complexity for target datasets in each task, which has a dependence of the feature variance V. If V is small, the order becomes $O(1/N_S)$ instead of $O(1/\sqrt{N_S})$, a significant drop for the number of the training datasets. In Section 4, we show that this result inspires us to include feature variance reduction in the synergistic learning process, which would significantly reduce using `MQUERY`.

Algorithm 1 Synergistic Learning

```
1: Given: model set $H_{\text{SynLearn}}$, data providers $\mathcal{D}$.
2: if Feature Pre-Training then
3:     repeat
4:         $L_{decom}(\mathcal{Z}_d) \leftarrow$ SQUERY($\mathcal{Z}_d$), $d = 1, \ldots, D$; UPDATE($g_\mu$; $\sum_{d=1}^{D} L_{decom}(\mathcal{Z}_d)$);
5:     until reach end.
6: end if
7: repeat
8:     Task & Model Sampling: $\mu_1, \mu_2 \ldots, \mu_t \sim \mathcal{T}, h_1, h_2 \ldots, h_m \sim \mathcal{M}$;
9:     Reusability Validation: $g(\mu_i, \mathcal{Z}_d, h_k) \leftarrow$ SQUERY($\mathcal{Z}_d$),
       $v^*(\mu_i, h_k) \leftarrow$ COMBINE$[\{\psi(\mu_i, \mathcal{Z}_d, h_k)\}_{d=1}^{|\bar{\mathcal{Z}}_{\mu_i}|}]$, $i \in [t], k \in [m], d \in [\bar{\mathcal{Z}}_{\mu_i}]$;
10:    Target Dataset Generation: $S_{\mu_i} \leftarrow$ MQUERY($\bar{\mathcal{Z}}_{\mu_i}$), $i = 1, \ldots, t$;
11:    UPDATE($\hat{v}$; $\{S_{\mu_i}\}_{i=1}^{t}, \{h_k\}_{k=1}^{m}, \{v^*(\mu_i, h_k)\}_{i,k=1}^{t,m}$) to minimize Equation 2;
12: until reach end.
```

4 Learning Method

In this section, we introduce the synergistic learning algorithm and the MRE model.

4.1 Synergistic Learning Algorithm

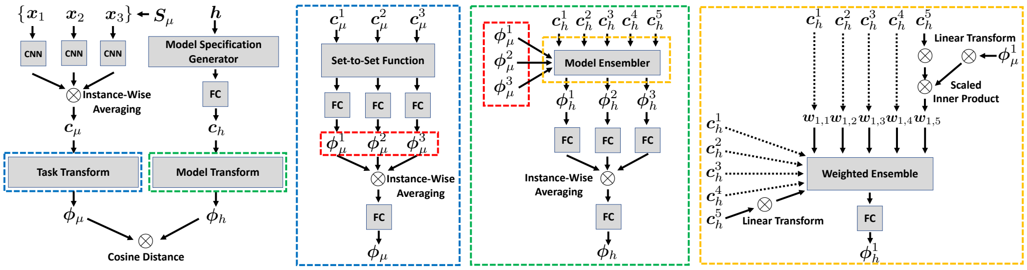
The realization of synergistic learning is illustrated in Algorithm 1. In this section, besides the metric learning step of minimizing Equation 2 (Line 11), we discuss other crucial steps below in general. We consider two general synergistic learning settings. One is named *isolated closure setting*, which indicates that any task closure includes a single data provider. Otherwise, we name it *grouped closure setting*, which indicates that there exist multi-data-provider task closures.

Task and model sampling (Line 8). To generate the training data for synergistic learning, tasks and models are needed be sampled. At this stage, no interaction to the data providers is needed.

Reusability validation (Line 9). The objective for reusability validation is to acquire value of v^* between any pairs of task and model that are sampled. In general, this can be done by sending models to the data providers and estimate v^* using v_{valid}. Under the isolated closure setting, only `SQUERY` is needed obviously. We could ask the data providers to use sufficiently large data, making v_{valid} an accurate estimator. But the key challenge appears under the grouped closure setting since the statistics used for calculating v_{valid} should be obtained from multiple providers at the same time. To achieve this by `SQUERY`, we require v_{valid} to have a special structure. Specifically, for task μ, let $S_\mu(\mathcal{Z}_d)$ be a sample generated from the marginal distribution of μ over $\mathcal{Z}_d$, a member of its task closure. We require $v_{valid}(\mu, h) = \texttt{COMBINE}[\{\psi(\mu, \mathcal{Z}_d, h)\}_{d=1}^{|\bar{\mathcal{Z}}_\mu|}]$, in which $\psi(\mu, \mathcal{Z}_d, h) = \mathbb{E}_{S_\mu(\mathcal{Z}_d) \sim \mu(\mathcal{Z}_d)}[\psi(S_\mu(\mathcal{Z}_d), h)]$. In this construction, $\psi(S_\mu(\mathcal{Z}_d), h)$ is a function over single data providers and `COMBINE` is an aggregation function decided by specific MRE function, thus only `SQUERY` is needed for calculation.

Target dataset generation (Line 10). For isolated closure setting, this step can be done by sampling from a data provider using `SQUERY`. However, for grouped closure setting, `MQUERY` is necessary for this step since the target datasets must include the raw data. This issue can be relieved by the insight brought from Theorem 3.1: For each task, the number of the target data needed is closely related to the data variance. A feature pre-training stage can be introduced to reduce using `MQUERY`.

Feature pre-training (Line 2-6). Feature pre-training is the preparation stage for synergistic learning. It is participated by all data providers, aiming at learning a feature extractor g_μ which could output low-variance features for the raw data. Once g_μ is learned, it would be integrated into the MRE model. As a result, in synergstic learning, the data providers only need to use much fewer data for answering `MQUERY`. On the other hand, the feature pre-training stage itself should be done by using only `SQUERY`. We argue that this can be realized by using learning objectives which are *decomposable w.r.t. the data providers*. More specifically, a decomposable objective is the summation of individual objectives, such that each individual objective L_{decom}^{d} only takes the data from a single


$\{x_1$ x_2 $x_3\}$ ← S_μ
h
CNN
CNN
CNN
Model Specification Generator
Instance-Wise Averaging
c_μ
FC
c_h
Task Transform
Model Transform
ϕ_μ
ϕ_h
Cosine Distance
c_μ^1 c_μ^2 c_μ^3
Set-to-Set Function
FC
ϕ_μ^1 ϕ_μ^2 ϕ_μ^3
Instance-Wise Averaging
c_h^1 c_h^2 c_h^3 c_h^4 c_h^5
Model Ensembler
ϕ_h^1 ϕ_h^2 ϕ_h^3
ϕ_h
Linear Transform
Scaled Inner Product
$w_{1,1}$ $w_{1,2}$ $w_{1,3}$ $w_{1,4}$ $w_{1,5}$
Weighted Ensemble
ϕ_h^1

(a) Basic Model (b) Task TM (c) Model TM (d) Model Ensembler

Figure 2: The models proposed for synergistic learning. (a) illustrates the basic model. (b) and (c) illustrate the task and model transform modules (TMs) used in classification. (d) illustrates the model ensembler. The components within the dashed box of the same color are the same.

data provider $\mathcal{Z}_d$ as its input. Then the purpose of the queries reduces to calculating each L^d_{decom}, which can be done by `SQUERY` as expected.

Due to space limitation, we provide more details on how to implement the above synergistic learning steps to different learning scenarios in Section B.

4.2 MRE Model

First, we introduce the basic structure of the MRE model defined in Section 3, which is illustrated in (a) of Figure 2. In this model, the target data is first processed by the task feature backbone to generate the feature c_μ. Afterwards, a task transform module is responsible to generate the task representation ϕ_μ which aggregates all the information from the target dataset. For isolated closure setting, the task transform module can be simply an instance-wise averaging operation. In correspondence, before the model feature backbone, a model specification generator is introduced which can be any function that transforms a model into a vector. The model feature c_h is then transformed into the model representation ϕ_h using the model transform module. For isolated closure setting, the model transform module can usually be omitted, i.e. $c_h = \phi_h$.

For many learning scenarios, using more elaborate MRE model structure is useful, such as classification. Below we consider small-data transfer learning for classification with supervised pre-trained models. A target task μ has L_T classes $\{y_l\}_{l=1}^{L_T}$ to distinguish, in which all the classes belong to the task class universe $\mathcal{Y}_T$. Meanwhile, for a pre-trained L_M-way classification model h, we denote by $\{\hat{y}_{l'}\}_{l'=1}^{L_M}$ the classes that the models are trained on. We assume that all the model classes belong to the model class universe $\mathcal{Y}_M$, which could be different from $\mathcal{Y}_T$. Similar to [Tran et al., 2019, Nguyen et al., 2020, You et al., 2021], we assume that any transfer strategy for this scenario, such as global fine-tuning and head re-training, can be used. While according to our experience, head re-training, in which the feature backbones of the pre-trained models are frozen and only the prediction heads are re-trained, is more proper than global fine-tuning for small-data transfer learning.

The overall structure of the MRE model for classification follows the basic model, but the task and model transform modules have more complex structures. For the task transform module ((b) of Figure 2), we take care of modeling the relationship among task classes. On one hand, instead of directly generating the feature representation of the whole task ϕ_μ, the module generates the features $\{\phi^l_\mu\}_{l=1}^{L_T}$ for all task classes first, and then aggregate them to form ϕ_μ. Any set-to-set transform module, such as Deep Sets [Zaheer et al., 2017] and Transformer [Vaswani et al., 2017, Ye et al., 2020], can be used. What is essential here is to generate the feature for one task class based on the context information from all other task classes.

More importantly, we propose an attention-based model transform module. This module is motivated by our observations during studying MRE: there is a close connection between $p(y_l, \hat{y}_{l'})$ and model reusability for classification, which is also pointed out by recent studies [Tran et al., 2019, Nguyen et al., 2020]. The module is illustrated in (c) of Figure 2. First, the module takes both the model and task class features $\{c^{l'}_h\}_{l'=1}^{L_M}, \{c^l_\mu\}_{l=1}^{L_T}$ as its inputs and transform them into the *model-attention-aware task class features* $\{\phi^l_h\}_{l=1}^{L_T}$. Subsequently, $\{\phi^l_h\}_{l=1}^{L_T}$ are aggregated to form the model feature ϕ_h. The core component of this module is the model ensembler ((d) of Figure 2). For a pair of model and task classes $y_l, \hat{y}_{l'}$, the model ensembler calculates the attention weight of y_l to $\hat{y}_{l'}$, which is $w_{l,l'}$, with the inner-product attention. And then, the model class features are linearly combined by the attention weights to form ϕ_h^l which represents the selected model class information on y_l. We require $w_{l,l'}$ to be closely related to $p(y_l, \hat{y}_{l'})$ to make it represent the correlation between y_l and $\hat{y}_{l'}$. Therefore, besides the metric loss defined in Equation 2, we introduce an additional *attention supervision loss* to supervise the learning of attention weights. For $w_{l,l'}$, we treat it as the output probability of a binary classifier. The training labels are generated from $p(y_l, \hat{y}_{l'})$. To be specific, we set the label to be one if $p(y_l|\hat{y}_{l'}) > \gamma_1, p(\hat{y}_{l'}|y_l) > \gamma_2$ and zero otherwise, in which γ_1, γ_2 are two thresholds. And then, the attention supervision loss is calculated from the logistic loss. Finally, the overall training loss is formulated as

$$L_{all} = L_{metric} + w_{att} L_{att}, \tag{3}$$

in which L_{metric} is the small-batch version of the metric loss defined in Equation 2, L_{att} is the attention supervision loss defined above, and w_{att} is the weight for L_{att}.

5 Experiments

In the experiments, we first do metric visualization to verify whether synergistic learning can learn meaningful metric space. Furthermore, we conduct experiments for both in-dataset and cross-dataset MRE to verify the performance of synergistic learning. All experiments are conducted on servers with NVIDIA Tesla V100 GPUs. The code[4] is implemented with TensorFlow [Abadi et al., 2016] (Apache 2.0 License). More details of the experimental setups are discussed in Section C and more experimental results are included in Section D.

5.1 Metric Visualization

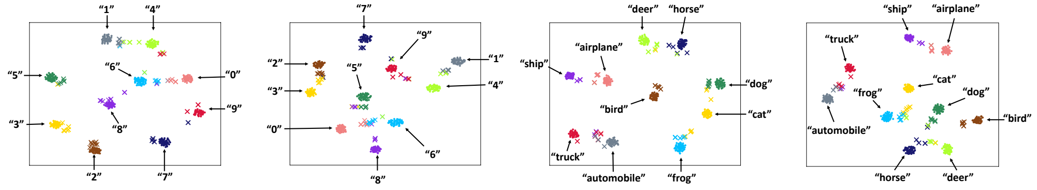
We adopt two ten-class datasets MNIST [LeCun et al., 1998] and CIFAR-10 [Krizhevsky, 2009] to visualize the task-model metric learned by synergistic learning. For each dataset, we randomly generated 20 five-class pre-trained models, as well as 20 data providers with the same class assignments, for synergistic learning, and another 20 models for testing. We treat each model as five detectors for the corresponding classes and try to use synergistic learning to obtain the detector-data metric. We consider two settings of synergistic learning. The first is learning with full data: all training sets are used for metric learning and there is no feature pre-training stage. The second is learning with part data: 10% of the training set for each class is used for metric learning, and there is a feature pre-training stage involved using the full training set. Figure 3 illustrates the t-SNE [Maaten and Hinton, 2008] visualizations of the learned metric, which is calculated from the testing models and instances. We can see that meaningful metric distance spaces are learned for both learning with full and part data: synergistic learning makes the instances and detectors with the same classes closer. These results provide preliminary proofs for the effectiveness of synergistic learning.

5.2 In-Dataset MRE

Next, we use the dSprites dataset [Matthey et al., 2017] for testing in-dataset MRE. dSprites consists of images generated from six latent factors. We select three factors, shape, scale, orientation, to form the task domains. Another two real-valued factors, position X and Y, are used as the prediction targets. There are 720 domains and 1024 instances under each domain in total. For each domain, we randomly select 800 instances as the training set which are used to obtain 720 pre-trained models. The remaining 224 instances are used as the testing set. We randomly select 503 domains as the in-distribution domains and the remaining 217 domains as the out-distribution domains. Under this setting, we treat each in-distribution domain as a data provider, thus there are 503 data providers in total. For synergistic learning, we only use the training sets and models from the in-distribution domains. $v_{valid}(\mu, h)$ is set according to the mean squared error (MSE). We set training $K = 10$, but testing $K = 1$ for verifying the performance under the extremely challenging situation. During testing, we generate testing tasks using the testing sets from either the in-distribution domains or the out-distribution domains. For in-distribution domains, the models used for testing consist of only

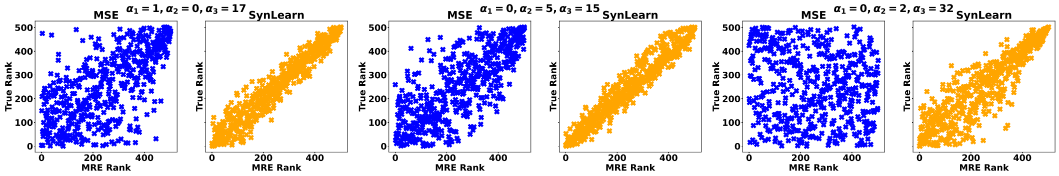
---

[4] The code is available on `https://github.com/candytalking/SynLearn`.

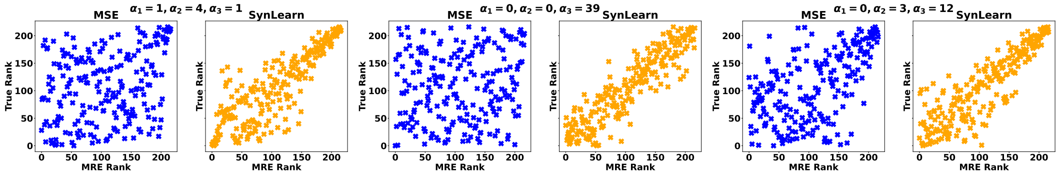


(a) MNIST (full data) (b) MNIST (part data) (c) CIFAR-10 (full data) (d) CIFAR-10 (part data)

Figure 3: t-SNE visualization of the learned metric on MNIST and CIFAR-10. (a) and (c) show the results learned with full data. (b) and (d) show the results learned with feature pre-training and part data. The class names for the data clusters (dot markers) are annotated. The detection models (cross markers) have the same color to their corresponding classes (better view in color mode).



(a) In-distribution domain 1 (b) In-distribution domain 2 (c) In-distribution domain 3



(d) Out-distribution domain 1 (e) Out-distribution domain 2 (f) Out-distribution domain 3

Figure 4: Performance comparison on dSprites: the MRE score rank vs. the true performance rank for MSE (left subfigure) and synergistic learning (right subfigure). (a-c) show in-distribution results on three random domains, and (d-f) show out-distribution results accordingly. The subfigure titles are the indexes of the domain factors. Please refer to Section D for more results.

in-distribution models. The out-distribution domains follow the similar rule. Figure 4 shows the ranking performance comparison between using MSE directly on the target datasets and synergistic learning (SynLearn). It can be observed that the synergistic learning ourperforms the naive MSE prediction significantly: under the extreme situation where testing $K = 1$, MSE almost fails totally in generating useful rankings, while synergistic learning can still perform desirably.

5.3 Cross-Dataset MRE

Finally, we use CIFAR-100 [Krizhevsky, 2009] and MiniImageNet [Vinyals et al., 2016] for cross-dataset MRE experiments. Results on more datasets are included in Section D. We consider two settings for experiments: Reuse CIFAR-100 pre-trained models on MiniImageNet target tasks and the opposite. For each setting, we pre-train 200 20-class models on the source dataset, in which 100 models are used for synergistic learning and the other 100 for testing. We use head re-training as the transfer strategy. For the target dataset, its pre-defined training set is used for synergistic learning and its pre-defined testing set is used for testing. All training and testing tasks are fixed to be five-way classification. For synergistic learning, we randomly generate 100 data providers, each of which holds the data of five target dataset classes. Full training set is used for metric pre-training and 10% of the training set is used for metric learning. For each testing task, 50 instances are sampled from each class to test accuracy. For performance evaluation, we use Kendall's τ-coefficient [ken] to measure the rank correlation between the MRE scores and the testing accuracy. For emphasizing the performance on top-performed models, we also employ the weighted version of Kendall's τ-coefficient, τ_w, for which an exchange between elements with rank r and s (starting from zero) has weight $1/(r+1)+1/(s+1)$ [Vigna, 2015]. We compare synergistic learning (SynLearn) with three state-of-the-art MRE methods: NCE [Tran et al., 2019], LEEP [Nguyen et al., 2020] and LogME

Table 1: Results for the classification experiments. For CIFAR-100 → MiniImageNet, the input shape is $32 \times 32 \times 3$. For MiniImageNet → CIFAR-100, the input shape is $84 \times 84 \times 3$. τ and τ_w indicate the Kendall's τ-coefficient and its weighted version calculated from 100 randomly generated five-class target tasks. The results are mean±95% confidence interval calculated from five random seeds. K indicates the number of training instances per class for each of the testing tasks.

Setting	Method	$K=5$		$K=10$		$K=15$		$K=20$	
		τ	τ_w	τ	τ_w	τ	τ_w	τ	τ_w
CIFAR-100 → MiniImageNet	LogME	0.139±0.070	0.180±0.187	0.200±0.079	0.359±0.231	0.243±0.056	0.420±0.217	0.255±0.042	0.418±0.270
	NCE	0.197±0.020	0.393±0.129	0.300±0.015	0.568±0.097	0.365±0.015	0.645±0.076	0.397±0.008	0.661±0.057
	LEEP	0.282±0.032	0.532±0.096	0.367±0.016	0.649±0.064	0.417±0.017	0.700±0.051	0.441±0.011	0.703±0.046
	SynLearn	**0.459±0.022**	**0.714±0.032**	**0.482±0.018**	**0.735±0.016**	**0.497±0.018**	**0.740±0.016**	**0.502±0.011**	**0.750±0.021**
MiniImageNet → CIFAR-100	LogME	0.166±0.044	0.312±0.176	0.244±0.083	0.413±0.164	0.288±0.087	0.478±0.163	0.310±0.078	0.479±0.189
	NCE	0.173±0.009	0.382±0.105	0.257±0.020	0.496±0.054	0.306±0.014	0.556±0.081	0.345±0.017	0.585±0.076
	LEEP	0.235±0.009	0.471±0.065	0.323±0.023	0.559±0.066	0.362±0.019	0.619±0.063	0.394±0.022	0.631±0.076
	SynLearn	**0.419±0.033**	**0.620±0.102**	**0.426±0.027**	**0.642±0.102**	**0.428±0.019**	**0.638±0.076**	**0.431±0.022**	**0.639±0.073**

[You et al., 2021]. The results are illustrated in Table 1. It can be observed that synergistic learning significantly outperforms other methods for small-data MRE. Note that for results in Table 1, we fix $K = 10$ during synergistic learning, while we observe significantly more robust performance of SynLearn over other approaches when the testing K varies. We conduct ablation studies to verify the effectiveness of the attention supervision, the feature pre-training, and the choice of the training K. The results are provided in Section D.

6 Related Work

Direct approaches for MRE. The direct approaches are based on the statistics calculated on the target training datasets as the MRE scores. The direct approaches involve only simple statistics calculation, meanwhile no auxiliary information is used, thus are usually quite simple and efficient. The representative studies of the direct approaches are [Tran et al., 2019, Nguyen et al., 2020]. In [Tran et al., 2019], the negative conditional entropy (NCE) score is proposed, which is an information-theoretic quantity measuring the entropy for $p(y|\hat{y})$. In [Nguyen et al., 2020], the log expected empirical prediction (LEEP) score is proposed, which can be regarded as an improvement of NCE. The LEEP score also closely related to $p(y|\hat{y})$. But it uses the *soft* prediction probability in calculation, to take the place of the *hard* label assignment calculation in NCE. Thus LEEP uses more information of prediction uncertainty. However, as shown in our experiments, both NCE and LEEP suffer from significant performance degeneration for small-data MRE. This is not surprising since the statistics calculated from small data usually have higher variance.

Learning approaches for MRE. The learning approaches conducts learning for MRE. Similar to the existing specification-based approaches, the testing-stage models and tasks are used for learning. In comparison, no testing tasks and models are necessary for synergistic learning. The representative approaches are [You et al., 2021, Achille et al., 2019]. In [You et al., 2021], the logarithm of maximum evidence (LogME) approach is proposed. Different from NCE and LEEP which aim at finding the correlation between the source model predictions and the target outputs, LogME builds the correlation between source model features and the target outputs. Training using target data is necessary for LogME. The advantage of LogME is its wide applicability for different learning problems. While its performance degenerates more significantly than NCE and LEEP under the small-data scenario. This is likely caused by the necessity of learning on the target data which would lead to stronger over-fitting. In [Achille et al., 2019], the Model2Vec approach is proposed. Model2Vec uses metric learning to build task-model metric, which is similar to synergistic learning. But Model2Vec focuses on generating the metric distances for a fixed set of models, thus is a learning approach for MRE. In comparison, synergistic learning generates the task-model metric for future MRE problems in which the models are unseen during synergistic learning.

Few-shot learning. In few-shot learning (FSL) [Vinyals et al., 2016, Ha et al., 2017, Snell et al., 2017, Finn et al., 2017], a meta-model is learned to solve future small-data learning tasks. Usually, FSL does not assume to use pre-trained models except for a few cases [Chowdhury et al., 2021]. For MRE, its relationship to FSL is similar to that to transfer learning: MRE can serve as a good preparation step for any FSL task that is allowed to use a pool of pre-trained models.

7 Limitations and Future Work

In this work, we proposed the synergistic learning approach for the small-data model reusability evaluation (MRE) problem. In this section, we discuss the limitations of synergistic learning, as well as possible future research directions.

The task-independent assumption. Synergistic learning is based on the assumption that the model transferability has little relationship with the task-independent properties of the small sample. But in real problems, *bad samples* indeed exist whose task-independent properties affect transfer performance. We believe that fundamental limits exist for dealing with these samples, thus the relaxation of this assumption is challenging. Note that Theorem 3.1 does not rely on this assumption, thus synergistic learning can be done even when it is not held. While lacking this special property could possibly degenerate the performance. Exploring how to tackle this challenge would be an important future research topic.

Effectiveness on more learning scenarios. Due to the resource limitation, we only conduct experiments for supervised pre-trained models. In recent years, reusing unsupervised pre-trained models has been increasingly popular. MRE for unsupervised pre-trained models is a very meaningful topic, on which the research is still missing as far as we know. Note that synergistic learning does not restrict the type of pre-trained models: it can be used whenever a ground-truth MRE function is defined. Thus it can also work for unsupervised pre-trained models in principle. We plan to verify this point in our future researches. Furthermore, in this paper, we only use image datasets for experiments. Applying synergistic learning on application in other modalities is also interesting.

MRE in learnware. Learnware [Zhou and Tan, 2022] is a growing research topic about reusing pre-trained models accommodated in a learnware market, which holds various machine learning models submitted by developers all over the world, to enable future user, who knows nothing about the models in the learnware market in advance, no need to build their own machine learning models from scratch. One of the key ingredient of learnware is *model specification*, which enable the models to be efficiently and adequately identified and reused, given the constraint that neither the training data of model developers nor that of users are leaked to the market. Once a set of potentially helpful learnwares have been identified for user, there can be various ways to reuse them to help address users own task. This paper can be viewed as providing a new way to reuse the identified learnwares.

Acknowledgements

This research is supported by NSFC (61921006, 62206245) and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We would like to thank the anonymous reviewers for their contructive suggestions, as well as Han-Jia Ye, Ming Pang, and Lu Wang for helpful discussions and their thoughtful comments.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: A system for {Large-Scale} machine learning. In *USENIX symposium on operating systems design and implementation (OSDI)*, pages 265–283, 2016.

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 6430–6439, 2019.

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

Arkabandhu Chowdhury, Mingchao Jiang, Swarat Chaudhuri, and Chris Jermaine. Few-shot image classification: Just use a library of pre-trained feature extractors and a simple classifier. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 9445–9454, 2021.

Stéphan Clémençon, Gábor Lugosi, Nicolas Vayatis, et al. Ranking and empirical minimization of u-statistics. *The annals of statistics*, 36(2):844–874, 2008.

Yao-Xiang Ding and Zhi-Hua Zhou. Boosting-based reliable model reuse. In *Proceedings of the asian conference on machine learning (ACML)*, pages 145–160, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the international conference on machine learning (ICML)*, pages 1126–1135, 2017.

Dylan J Foster, Ayush Sekhari, and Karthik Sridharan. Uniform convergence of gradients for non-convex learning and optimization. In *Advances in neural information processing systems 31 (NeurIPS)*, pages 8745–8756, 2018.

Xiand Gao, Xiaobo Li, and Shuzhong Zhang. Online learning with non-convex losses and non-stationary regret. In *International conference on artificial intelligence and statistics (AISTATS)*, pages 235–243, 2018.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. In *International conference on learning representations (ICLR)*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset, 2017.

Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical programming*, 108(1):177–205, 2006.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *Proceedings of the international conference on machine learning (ICML)*, pages 7294–7305, 2020.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems 31 (NeurIPS)*, pages 719–729, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems 30 (NIPS)*, pages 4080–4090, 2017.

Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 1395–1405, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems 30 (NIPS)*, pages 6000–6010, 2017.

Sebastiano Vigna. A weighted correlation index for rankings with ties. In *Proceedings of the 24th international conference on World Wide Web*, pages 1166–1176, 2015.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems 29 (NIPS)*, pages 3630–3638, 2016.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Xi-Zhu Wu, Wenkai Xu, Song Liu, and Zhi-Hua Zhou. Model reuse with reduced kernel mean embedding specification. *arXiv preprint arXiv:2001.07135*, 2020.

Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 8808–8817, 2020.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *Proceedings of the international conference on machine learning (ICML)*, pages 12133–12143, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems 30 (NIPS)*, pages 3391–3401, 2017.

Zhi-Hua Zhou and Zhi-Hao Tan. Learnware: Small models do big. *arXiv preprint arXiv:2210.03647*, 2022.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in neural information processing systems 23 (NIPS)*, pages 2595–2603, 2010.

Checklist

1. For all authors...
    (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Abstract and Section 1.
    (b) Did you describe the limitations of your work? [Yes] See Section 7.
    (c) Did you discuss any potential negative societal impacts of your work? [No] We haven't found any potential negative societal impacts for this research.
    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the guidelines and checked our paper accordingly.
2. If you are including theoretical results...
    (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section A in the appendix.
    (b) Did you include complete proofs of all theoretical results? [Yes] See Section A in the appendix.
3. If you ran experiments...
    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 5.
    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 in the paper and Section C in the appendix.
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1.
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5 in the paper and Section D in the appendix.
    (b) Did you mention the license of the assets? [Yes] See Section 5.
    (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The full code is released online. The URL is on Page 7.
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use only open-source resources and publicly-available datasets.
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We do not use resources of this kind.
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not use crowdsourcing or conducted research with human subjects.
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not use crowdsourcing or conducted research with human subjects.

A Theoretical Results

In this section, we provide the full version of Theorem 3.1 and its proof.

A.1 Full Version of Theorem 3.1

Our analysis is based on the following assumptions about Θ.

Assumption A.1. *Θ satisfies the following conditions*[5]*:*

1. *Parameter boundness: $\forall\theta \in \Theta, \exists\beta_1 > 0, \|\theta\| \leq \beta_1$.*
2. *Gradient boundness: $\forall\theta \in \Theta, S_\mu, h, \exists\beta_2 > 0, \|\nabla_\theta[\hat{v}_\theta(S_\mu, h)]\| \leq \beta_2$.*
3. *Optimal parameter: $\exists\theta^* \in \Theta, \forall\theta \in \Theta, S_\mu, S_{\mu'}, h, h'$, if $\Delta_{v^*}(\mu, \mu', h, h') < 0$, then*
$$\Delta_{\hat{v}_{\theta^*}}(S_\mu, S_{\mu'}, h, h') \leq \Delta_{\hat{v}_\theta}(S_\mu, S_{\mu'}, h, h').$$
4. *Generalized star-convex property: $\forall\theta \in \Theta, S_\mu, S_{\mu'}, h, h', \exists\beta_3 \geq 1$,*
$$\Delta_{\hat{v}_\theta} - \Delta_{\hat{v}_{\theta^*}} \leq \beta_3\langle\nabla_\theta[\Delta_{\hat{v}_\theta}], \theta - \theta^*\rangle$$
in which $\Delta_{\hat{v}_\theta}$ is the shorthand for $\Delta_{\hat{v}_\theta}(S_\mu, S_{\mu'}, h, h')$, $\Delta_{\hat{v}_{\theta^*}}$ is the shorthand for $\Delta_{\hat{v}_{\theta^*}}(S_\mu, S_{\mu'}, h, h')$.

Remark. In Assumption A.1, the first two assumptions are common ones satisfied by most of the popular differentiable learning models. The third assumption states the existence of the optimal parameter in Θ. It is easy to see that $\theta^* \in \Theta^*$ in which $\Theta^* = \arg\min_{\theta\in\Theta} R^\gamma(\hat{v}_\theta)$. The fourth assumption is the generalization of the star-convex property Nesterov and Polyak [2006], Gao et al. [2018] which is a classic assumption in non-convex optimization theory.

Next, we propose the full version of Theorem 3.1.

Theorem A.2. *Under Assumption A.1, for any $\theta \in \Theta$, the following event holds with probability at least $1 - \delta$:*
$$R^\gamma(\theta) - R^\gamma(\theta^*) \leq 2\beta_1\beta_3\big[\nabla[\hat{R}(\bar{G}_\theta)] + \Delta(N_t, N_m, N_S)\big],$$
in which
$$\begin{aligned}\Delta(N_t, N_m, N_S) =& \frac{8(4\beta_2^2+1)}{\sqrt{\lfloor\frac{\min\{N_t,N_m\}}{2}\rfloor}} + 4\sqrt{2}\beta_2\Big(\frac{1}{\sqrt{N_t}} + \frac{1}{\sqrt{N_m}}\Big)\log\frac{4}{\delta} \\ &+ \frac{8\beta_2\log(8/\delta)}{\lfloor\frac{\min\{N_t,N_m\}}{2}\rfloor} + \frac{4\beta_2\log(2(N_tN_m)^2/\delta)}{3N_S} + \sqrt{\frac{2V\log(2(N_tN_m)^2/\delta)}{N_S}}.\end{aligned}$$

A.2 Proof of Theorem A.2

First, we intrdouce two concentration inequalities. The first is the well-known bounded difference concentration bound named McDiarmid inequality.

Lemma A.3 (McDiarmid inequality McDiarmid [1989]). *Assume that there is a real-valued function $f : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \mathcal{X}_m \mapsto \mathbb{R}$ such that for any $i \in [m]$ and $x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \ldots, x_i \in \mathcal{X}_i, \ldots, x_m \in \mathcal{X}_m, x_i' \in \mathcal{X}_i$, the following bounded derivative condition holds:*
$$\|f(x_1, x_2, \cdots, x_i, \cdots, x_m) - f(x_1, x_2, \cdots, x_i', \cdots, x_m)\| \leq c_i.$$
Then for m independent random variables X_i over $\mathcal{X}_i, i \in [m]$ and $\forall\epsilon > 0$,
$$\Pr(f(X_1, X_2, \cdots, X_m) - \mathbb{E}[f(X_1, X_2, \cdots, X_m)] \geq \epsilon) \leq \exp(-2\epsilon^2/\sum_{i=1}^m c_i^2),$$
$$\Pr(f(X_1, X_2, \cdots, X_m) - \mathbb{E}[f(X_1, X_2, \cdots, X_m)] \leq -\epsilon) \leq \exp(-2\epsilon^2/\sum_{i=1}^m c_i^2).$$

---

[5] We assume 2-norm by default.

The second is a Bernstein-type bounded difference inequality which provides a variance-dependent concentration bound. The proof can be found in Section A.1.3 of Cesa-Bianchi and Lugosi [2006].

Lemma A.4. *Assume that there is a real-valued function* $f : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \mathcal{X}_m \mapsto \mathbb{R}$ *such that for any* $i \in [m]$ *and* $x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \dots, x_i \in \mathcal{X}_i, \dots, x_m \in \mathcal{X}_m, x_i' \in \mathcal{X}_i$*, the following bounded derivative condition holds:*

$$\|f(x_1, x_2, \cdots, x_i, \cdots, x_m) - f(x_1, x_2, \cdots, x_i', \cdots, x_m)\| \le c_i.$$

For m *independent random variables* X_i *over* $\mathcal{X}_i, i \in [m]$*, let*

$$V_i = \mathbb{E}[f(X_1, X_2, \cdots, X_m)|X_1, X_2, \cdots, X_i] - \mathbb{E}[f(X_1, X_2, \cdots, X_m)|X_1, X_2, \cdots, X_{i-1}],$$
$$V = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}[V_i^2].$$

Then for $\forall \epsilon > 0$*,*

$$\Pr(f(X_1, X_2, \cdots, X_m) - \mathbb{E}[f(X_1, X_2, \cdots, X_m)] \ge \epsilon) \le \exp(-m^2\epsilon^2/[2(mV + \epsilon(\sum_{i=1}^{m} c_i)/3)]),$$
$$\Pr(f(X_1, X_2, \cdots, X_m) - \mathbb{E}[f(X_1, X_2, \cdots, X_m)] \le -\epsilon) \le \exp(-m^2\epsilon^2/[2(mV + \epsilon(\sum_{i=1}^{m} c_i)/3)]).$$

Our first task is to transform the access risk to the expectation of the gradient norm for the metric gap. This is done by the following key lemma.

Lemma A.5. *Let* $\nabla[\tilde{G}_\theta(\mu, \mu', h, h')] = \mathbb{E}_{\mu,\mu'}\Big[\mathbf{I}[\Delta_{v^*}(\mu, \mu', h, h') < 0]\nabla_\theta[\bar{G}_\theta(S_\mu, S_{\mu'}, h, h')]\Big]$*. For* $\forall \theta \in \Theta$*,*

$$R^\gamma(\hat{v}_\theta) - R^\gamma(\hat{v}_{\theta^*}) \le 2\beta_1\beta_3\|\mathbb{E}_{\mathcal{T},\mathcal{M}}\big[\nabla[\tilde{G}_\theta(\mu, \mu', h, h')]\big]\|.$$

Proof. First, we need to prove that $\forall \theta \in \Theta, S_\mu, S_{\mu'}, h, h'$ such that $\Delta_{v^*}(S_\mu, S_{\mu'}, h, h') < 0$,

$$r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_\theta) - r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_{\theta^*}) \le \Delta_{\hat{v}_\theta}(S_\mu, S_{\mu'}, h, h') - \Delta_{\hat{v}_{\theta^*}}(S_\mu, S_{\mu'}, h, h').$$

Case 1: $r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_\theta) = 0$. In this case, the result obviously holds by the optimality of θ^*.

Case 2: $r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_\theta) > 0, r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_{\theta^*}) \le 0$. In this case,

$$\begin{aligned} &r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_\theta) - r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_{\theta^*}) \\ \le&\Delta_{\hat{v}_\theta}(S_\mu, S_{\mu'}, h, h') + \gamma \le \Delta_{\hat{v}_\theta}(S_\mu, h, h') - \Delta_{\hat{v}_{\theta^*}}(S_\mu, h, h'). \end{aligned}$$

Case 3: $r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_\theta) > 0, r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_{\theta^*}) > 0$. We have

$$r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_\theta) - r^\gamma_{v^*}(S_\mu, S_{\mu'}, h, h'; \hat{v}_{\theta^*}) = \Delta_{\hat{v}_\theta}(S_\mu, S_{\mu'}, h, h') - \Delta_{\hat{v}_{\theta^*}}(S_\mu, S_{\mu'}, h, h').$$

As a result, we have that

$$\begin{aligned} &R^\gamma(\hat{v}_\theta) - R^\gamma(\hat{v}_{\theta^*}) \\ \le&\mathbb{E}_{\mathcal{T},\mathcal{M},\mu,\mu'}\Big[\mathbf{I}[\Delta_{v^*}(\mu, \mu', h, h') < 0](\Delta_{\hat{v}_\theta}(S_\mu, S_{\mu'}, h, h') - \Delta_{\hat{v}_{\theta^*}}(S_\mu, S_{\mu'}, h, h'))\Big] \\ \le&\beta_3\langle\mathbb{E}_{\mathcal{T},\mathcal{M},\mu,\mu'}\big[\mathbf{I}[\Delta_{v^*}(\mu, \mu', h, h') < 0]\nabla_\theta[\bar{G}_\theta(S_\mu, S_{\mu'}, h, h')]\big], \theta - \theta^*\rangle \\ \le&2\beta_1\beta_3\mathbb{E}_{\mathcal{T},\mathcal{M},\mu,\mu'}\Big[\|\mathbf{I}[\Delta_{v^*}(\mu, \mu', h, h') < 0]\nabla_\theta[\bar{G}_\theta(S_\mu, S_{\mu'}, h, h')]\|\Big] \\ =&2\beta_1\beta_3\mathbb{E}_{\mathcal{T},\mathcal{M}}\Big[\mathbb{E}_{\mu,\mu'}\big[\|\mathbf{I}[\Delta_{v^*}(\mu, \mu', h, h') < 0]\nabla_\theta[\bar{G}_\theta(S_\mu, S_{\mu'}, h, h')]\|\big]\Big], \end{aligned}$$

in which the second inequality is due to the star-convexity assumption and the third inequality is due to the Hölder inequality. □

The above lemma seperates the expectation over training datasets from tasks and models. Let us study the outer expectation first. This expectation is taken over complex dependencies among tasks and models. To disentangle the dependencies, we propose the tetrad block lemma which generalizes the pairwise block lemma Clémençon et al. [2008], a key result for analysing the U-statistics.

Lemma A.6. *Let $B_1 = \min\{Q, K\}$. For any function of tetrads $f_\theta(a, a', b, b')$ in which $a, a' \in \mathcal{A}, b, b' \in \mathcal{B}$, convex function ψ, and distributions over $\mathcal{A}, \mathcal{B}$,*

$$\begin{aligned}
&\mathbb{E}\sup_\theta[\psi(\frac{1}{Q(Q-1)K(K-1)}\sum_{q,q',k,k'=1}^{Q,Q-1,K,K-1} f_\theta(a_q, a_{q'}, b_k, b_{k'}))] \\
\leq &\mathbb{E}\sup_\theta[\psi(\frac{1}{\lfloor\frac{B_1}{2}\rfloor}\sum_{k=1}^{\lfloor\frac{B_1}{2}\rfloor} f_\theta(a_k, a_{k+\lfloor\frac{B_1}{2}\rfloor}, b_k, b_{k+\lfloor\frac{B_1}{2}\rfloor}))].
\end{aligned} \tag{4}$$

Proof. Assume that $Q \geq K$. Denote π_1 as an element of the K-permutations of $[Q]$ and π_2 as an element of the permutations of $[K]$. Furthermore, denote $\pi_1(k), \pi_2(k)$ as the k-th element of π_1 and π_2. We have

$$\begin{aligned}
&\mathbb{E}\sup_\theta[\psi(\frac{1}{(Q(Q-1)K(K-1)}\sum_{q,q',k,k'=1}^{Q,Q-1,K,K-1} f_\theta(a_q, a_{q'}, b_k, b_{k'}))] \\
=&\mathbb{E}\sup_\theta[\psi(\frac{1}{Q(Q-1)K(K-1)}\sum_{\pi_1,\pi_2}\frac{1}{\lfloor\frac{K}{2}\rfloor}\sum_{k=1}^{\lfloor\frac{K}{2}\rfloor} f_\theta(a_{\pi_1(k)}, a_{\pi_1(k+\lfloor\frac{K}{2}\rfloor)}, b_{\pi_2(k)}, b_{\pi_2(k+\lfloor\frac{K}{2}\rfloor)}))] \\
\leq&\mathbb{E}\sup_\theta\frac{1}{Q(Q-1)K(K-1)}\sum_{\pi_1,\pi_2}[\psi(\frac{1}{\lfloor\frac{K}{2}\rfloor}\sum_{k=1}^{\lfloor\frac{K}{2}\rfloor} f_\theta(a_{\pi_1(k)}, a_{\pi_1(k+\lfloor\frac{K}{2}\rfloor)}, b_{\pi_2(k)}, b_{\pi_2(k+\lfloor\frac{K}{2}\rfloor)}))] \\
\leq&\frac{1}{Q(Q-1)K(K-1)}\sum_{\pi_1,\pi_2}\mathbb{E}\sup_\theta[\psi(\frac{1}{\lfloor\frac{K}{2}\rfloor}\sum_{k=1}^{\lfloor\frac{K}{2}\rfloor} f_\theta(a_{\pi_1(k)}, a_{\pi_1(k+\lfloor\frac{K}{2}\rfloor)}, b_{\pi_2(k)}, b_{\pi_2(k+\lfloor\frac{K}{2}\rfloor)}))] \\
=&\mathbb{E}\sup_\theta[\psi(\frac{1}{\lfloor\frac{K}{2}\rfloor}\sum_{k=1}^{\lfloor\frac{K}{2}\rfloor} f_\theta(a_k, a_{k+\lfloor\frac{K}{2}\rfloor}, b_k, b_{k+\lfloor\frac{K}{2}\rfloor}))],
\end{aligned}$$

in which the first inequality is obtained from the convexity of ψ, and the second inequaility is obtained from the property of the sup function as well as the linearity of expectation. The same proof is valid for $Q < K$. ☐

Note that comparing to the pairwise block lemma, in the above lemma, ψ has a different position in the inequality. This fits for our need below. As a result, we do not need to assume that ψ is non-decreasing as in the pairwise block lemma.

According to Lemma A.5, we should upper bound $\|\mathbb{E}_{\mathcal{T},\mathcal{M}}\big[\nabla[\tilde{G}_\theta(\mu, \mu', h, h')]\big]\|$. Below we denote $\mathbb{E}[\nabla\tilde{G}_\theta] = \mathbb{E}_{\mathcal{T},\mathcal{M}}\big[\nabla[\tilde{G}_\theta(\mu, \mu', h, h')]\big]$ for simplicity. Furthermore, define $\|\nabla\hat{G}_\theta\|$ as

$$\|\frac{1}{(N_t N_m)^2}\sum_{i,i',k,k'=1}^{N_t,N_t,N_m,N_m}\mathbb{E}_{\mu,\mu'}\Big[\mathbf{I}[\Delta_{v^*}(\mu_i, \mu_{i'}, h_k, h_{k'}) < 0]\nabla\bar{G}_\theta(S_{\mu_i}, S_{\mu_{i'}}, h_k, h_{k'})\Big]\|.$$

Subsequently, we have

$$\|\mathbb{E}[\nabla\tilde{G}_\theta]\| \leq \|\nabla\hat{G}_\theta\| + \sup_{\theta\in\Theta}\|\mathbb{E}[\nabla\tilde{G}_\theta] - \nabla\hat{G}_\theta\|.$$

$\sup_{\theta\in\Theta}\|\mathbb{E}[\nabla\tilde{G}_\theta] - \nabla\hat{G}_\theta\|$ is the function of $N_t + N_m$ i.i.d. random variables $\{\mu_i\}_{i=1}^{N_t}, \{h_k\}_{k=1}^{N_m}$. Meanwhile, $\{\mu_i\}_{i=1}^{N_t}$ satisfy the bounded difference condition with parameter $8\beta_2/N_t$ and $\{h_k\}_{k=1}^{N_m}$

satisfy the bounded difference condition with parameter $8\beta_2/N_m$. As a result, according to Lemma A.3, we have that

$$\|\mathbb{E}[\nabla\tilde{G}_\theta]\| \le \|\nabla\hat{G}_\theta\| + \mathbb{E}\sup_{\theta\in\Theta}\|\mathbb{E}[\nabla\tilde{G}_\theta] - \nabla\hat{G}_\theta\| + 4\sqrt{2}\beta_2(\frac{1}{\sqrt{N_t}} + \frac{1}{\sqrt{N_m}})\log\frac{4}{\delta}$$

holds with probability at least $1-\delta/4$. The following task is to upper bound $\mathbb{E}\sup_{\theta\in\Theta}\|\mathbb{E}[\nabla\tilde{G}_\theta] - \nabla\hat{G}_\theta\|$. Let $N_{\min} = \llcorner\frac{\min\{N_t,N_m\}}{2}\lrcorner$. According to Lemma A.6,

$$\mathbb{E}\sup_{\theta\in\Theta}\|\mathbb{E}[\nabla\tilde{G}_\theta] - \nabla\hat{G}_\theta\| \le \mathbb{E}\sup_{\theta\in\Theta}\|\frac{1}{N_{\min}}\sum_{k=1}^{N_{\min}}\mathbb{E}[\nabla\tilde{G}_\theta] - \nabla\tilde{G}_\theta(\mu_k, \mu_{k+N_{\min}}, h_k, h_{k+N_{\min}})\|.$$

We introduce two useful technical tools for bounding the Rademacher complexity of gradient norms, which are proposed in Foster et al. [2018]. The proofs of these lemmas can be found therein.

Lemma A.7 (Symmetrization Lemma for Rademacher Complexity. Proposition 2 of Foster et al. [2018])**.** *Assume that $L_\theta(z)$ is a real-valued function over $z \in \mathcal{Z}$ parameterized by $\theta \in \Theta$. Furthermore, assume that the gradient norm of $L_\theta(z)$ is bounded: $\forall z \in \mathcal{Z}, \|\nabla L_\theta(z)\| \le \beta$. Then $\forall\delta > 0$, the following result holds with probability at least $1-\delta$ over random draw of i.i.d. sample $Z_N = \{z_n\}_{n=1}^N$ from a distribution $\mathcal{D}$ over $\mathcal{Z}$,*

$$\mathbb{E}_{Z_N\sim\mathcal{D}}\big[\sup_{\theta\in\Theta}\|\mathbb{E}_{z\sim\mathcal{D}}[\nabla L_\theta(z)] - \frac{1}{N}\sum_{n=1}^N\nabla L_\theta(z_n)\|\big] \le \frac{4}{N}\mathbb{E}_{\epsilon_n}\sup_{\theta\in\Theta}\|\sum_{n=1}^N\epsilon_n\nabla L_\theta(z_n)\| + \frac{4\beta\log\frac{2}{\delta}}{N},$$

in which $\{\epsilon_n\}_{n=1}^N$ are Rademacher random variables.

Lemma A.8 (Chain Rule for Rademacher Complexity. Theorem 1 of Foster et al. [2018])**.** *Given function $G:\mathbb{R}\to\mathbb{R}$ and a sequence of real-valued functions $F_n, n\in[N]$, assume that there are constants L_G and L_F such that $\|\nabla G\| \le L_G$ and $\forall n\in[N], \|\nabla F_n\| \le L_F$. Then the following result holds:*

$$\mathbb{E}_{\epsilon_n}\sup_{\theta\in\Theta}\|\sum_{n=1}^N\epsilon_n\nabla G(F_n)\| \le 2L_F\mathbb{E}_{\epsilon_n}\sup_{\theta\in\Theta}\sum_{n=1}^N\langle\epsilon_n, \nabla G(F_n)\rangle + 2L_G\mathbb{E}_{\epsilon_n}\sup_{\theta\in\Theta}\|\sum_{n=1}^N\langle\epsilon_n, \nabla F_n\rangle\|,$$

in which $\{\epsilon_n\}_{n=1}^N$ are Rademacher random variables, the derivatives in the term of the left-hand side and the second term of the right-hand side are taken over F_n, and the derivatives in the first term of the right-hand side are taken over G.

Using Lemma A.7, we have

$$\begin{aligned}&\mathbb{E}\sup_{\theta\in\Theta}\|\frac{1}{N_{\min}}\sum_{k=1}^{N_{\min}}\mathbb{E}[\nabla\tilde{G}_\theta] - \nabla\tilde{G}_\theta(\mu_k, \mu_{k+N_{\min}}, h_k, h_{k+N_{\min}})\|\\ \le&\frac{4}{N_{\min}}\mathbb{E}_{\epsilon_k}\sup_{\theta\in\Theta}\|\sum_{k=1}^{N_{\min}}\epsilon_k\nabla\tilde{G}_\theta(\mu_k, \mu_{k+N_{\min}}, h_k, h_{k+N_{\min}})\| + \frac{8\beta_2\log\frac{8}{\delta}}{N_{\min}}\end{aligned}$$

holds with probability at least $1-\delta/4$. Let $\Delta_\theta(\mu,\mu',h,h') = \mathbb{E}_{\mu,\mu'}[\mathbf{I}[\Delta_{v^*}(S_\mu, S_{\mu'}, h, h') < 0]\Delta_\theta(S_\mu, S_{\mu'}, h, h')]$. According to Lemma A.8, we have

$$\begin{aligned}&\mathbb{E}_\epsilon\sup_{\theta\in\Theta}\|\sum_{k=1}^{\llcorner\frac{\min\{N_t,N_m\}}{2}\lrcorner}\epsilon_k\nabla\tilde{G}_\theta(\mu_k, \mu_{k+N_{\min}}, h_k, h_{k+N_{\min}})\|\\ \le&4\beta_2\mathbb{E}_{\epsilon_k}\sup_{\theta\in\Theta}\big[\sum_{k=1}^{N_{\min}}\langle\epsilon_k, \nabla[\Delta_\theta(\mu_k, \mu_{k+N_{\min}}, h_k, h_{k+N_{\min}})]\rangle\big] + 2\mathbb{E}_{\epsilon_k}\|\sum_{k=1}^{N_{\min}}\epsilon_k\|\\ \le&2(4\beta_2^2+1)\mathbb{E}_{\epsilon_k}\|\sum_{k=1}^{N_{\min}}\epsilon_k\|,\end{aligned}$$

in which the second inequality is from the Cauchy-Schwarz inequality. Furthermore, by the convexity of $y = x^2$, we have

$$\mathbb{E}_\epsilon\|\sum_{k=1}^{N_{\min}}\epsilon_k\| \le [\mathbb{E}_\epsilon[\|\sum_{k=1}^{N_{\min}}\epsilon_k\|^2]]^{1/2} = \sqrt{N_{\min}}.$$

Combine the above results, we have with probability at least $1 - \delta/2$,

$$
\begin{aligned}
&\|\mathbb{E}_{\mathcal{T},\mathcal{M}}\big[\nabla[\tilde{G}_\theta(\mu, h, h')]\big]\| \\
\leq &\|\nabla \hat{G}_\theta\| + \frac{8(4\beta_2^2 + 1)}{\sqrt{\lfloor \frac{\min\{N_t, N_m\}}{2} \rfloor}} + 4\sqrt{2}\beta_2(\frac{1}{\sqrt{N_t}} + \frac{1}{\sqrt{N_m}}) \log \frac{4}{\delta} + \frac{8\beta_2 \log \frac{8}{\delta}}{\lfloor \frac{\min\{N_t, N_m\}}{2} \rfloor}.
\end{aligned}
$$

Now the task becomes to bound $\|\nabla \hat{G}_\theta\|$. We have that

$$
\begin{aligned}
\|\nabla \hat{G}_\theta\| =& \|\frac{1}{(N_t N_m)^2} \sum_{i,i',k,k'=1}^{N_t,N_t,N_m,N_m} \mathbb{E}_\mu \Big[\mathbf{I}[\Delta_{v^*}(\mu_i, \mu_{i'}, h_k, h_{k'}) < 0] \nabla \bar{G}_\theta(S_{\mu_i}, S_{\mu_{i'}}, h_k, h_{k'})\Big] \|. \\
\leq& \frac{1}{(N_t N_m)^2} \sum_{i,i',k,k'=1}^{N_t,N_t,N_m,N_m} \mathbb{E}_\mu \Big[\|\mathbf{I}[\Delta_{v^*}(\mu_i, \mu_{i'}, h_k, h_{k'}) < 0] \nabla \bar{G}_\theta(S_{\mu_i}, S_{\mu_{i'}}, h_k, h_{k'})\|\Big].
\end{aligned}
$$

When $\mu_i, \mu_{i'}, h_k, h_{k'}$ are fixed, we have the observation that $\frac{1}{N_S^2} \sum_{j,j'} \|\mathbf{I}[\Delta_{v^*}(\mu_i, \mu_{i'}, h_k, h_{k'}) < 0] \nabla \bar{G}_\theta(S_{\mu_i,j}, S_{\mu_{i'},j'}, h_k, h_{k'})\|$ is the function of N_S i.i.d. samples when $i = i'$ or function of $2N_S$ i.i.d. samples when $i \neq i'$, both satisfying the bounded difference property with parameter $4\beta_2/N_S$. By Lemma A.4, we have that

$$
\begin{aligned}
&\|\nabla \hat{G}_\theta\| \\
\leq &\frac{1}{(N_t N_m N_S)^2} \sum_{i,i',k,k'=1}^{N_t,N_t,N_m,N_m} \Big[\sum_{j,j'=1}^{N_S,N_S} \|\mathbf{I}[\Delta_{v^*}(\mu_i, \mu_{i'}, h_k, h_{k'}) < 0] \nabla \bar{G}_\theta(S_{\mu_i,j}, S_{\mu_{i'},j'}, h_k, h_{k'})\| \\
&+ \frac{4\beta_2 \log(2(N_t N_m)^2/\delta)}{3N_S} + \sqrt{\frac{2V \log(2(N_t N_m)^2/\delta)}{N_S}} \Big]
\end{aligned}
$$

holds with probability at least $1 - \delta/2$. Then we finally arrive at the access risk bound in the theorem.

B Realization of Synergistic Learning

In this section, we discuss more details about the realization of the crucial steps of synergistic learning. In special, we focus on the steps of reusability validation and feature pre-training.

B.1 Isolated Closure Setting

Isolated closure setting assumes that the task closure will always include only a single data provider. *We assume isolated closure setting in our dSprites experiment, which is under the scenario of regression.* While the algorithm can be used on general learning scenarios, such as classification, metric learning, ranking, etc, when the isolated closure assumption is satisfied.

Target dataset generation and feature pre-training. The target data can be generated easily with `SQUERY` from the single data provider within the task closure. Thus no feature pre-training is needed.

Reusability validation. The reusability valiation step requires to do MRE with v_{valid}. There are two basic modes. One is the *direct predict mode*, which can be used when the pre-trained models are directly used without fine-tuning in the target tasks. Another is the *fine-tune mode*, which can be used when the pre-trained models should be fine-tuned in the target tasks. For both modes, only `SQUERY` is needed for the isolated setting. For direct predict mode, the pre-trained model can be sent to the data provider, and then the prediction performance can be tested using arbitrarily large data. This is the situation in the dSprites experiment, in which the mean-squared error (MSE) is used directly for calculating v_{valid}. For the fine-tune mode, the pre-trained model can also be sent to the data provider, and the provider uses the model to do fine-tuning and testing using her own data. Finally, the testing performance can be returned using `SQUERY`. Therefore, there is naturally no need to use `MQUERY` for reusability validation under the isolated closure setting.

B.2 Grouped Closure Setting

Under the grouped closure setting, there are multi-data-provider task closure exist.*We assume the grouped closure setting in our cross-dataset experiments, which are classification tasks.* The algorithm can also be applied on general learning scenarios under the grouped closure assumption.

Target dataset generation and feature pre-training. Under the grouped closure setting, to generate a target training dataset for classification, `MQUERY` is necessary. Thus the feature pre-training stage should be introduced for learning g_μ to output low-variance features, then the data providers can use much fewer data to answer `MQUERY` according to Theorem 3.1. Any algorithm capable to reduce feature variance, such as self-supervised learning and metric learning, can be used for learning g_μ when the objective satisfies the decomposable condition discussed in Section 4. In the cross-dataset classification experiments, we introduce the following metric-based approach. We assume that the task closures satisfy the following conditions: each data provider $\mathcal{Z}_d$ covers a subset of $\mathcal{Y}_T^d \subseteq \mathcal{Y}_T$ such that $|\mathcal{Y}_T^d| > 1$. We define decomposable objective L_{decom} as follows. For data provider $\mathcal{Z}_d$, she samples a pair of classes y, y' from $\mathcal{Y}_T^d$ as well as two samples $\{S_i^y\}_{i=1}^N, \{S_i^{y'}\}_{i=1}^N$ from the two classes, in which $S_i^y, S_i^{y'}$ are training$K$-sized batches from class y, y'. And then she can calculate the following metric loss as L_{decom}^d:

$$\sum_{i,i'=1}^{N,N} \left[\|g_\mu(S_i^y) - g_\mu(S_{i'}^y)\| - \|g_\mu(S_i^y) - g_\mu(S_{i'}^{y'})\| + \gamma_{pre}\right]_+,$$

in which $\gamma_{pre} > 0$. This is a simple metric loss for reducing the intra-class distance and increasing the inter-class distance. By summing L_{decom}^d over all data providers, the objective can be used to reduce the global feature variance.

Reusability validation. For estimating $v^*(\mu, h)$ of a model h on task μ, a naive approach is to conduct model transfer and performance testing on h using a sufficiently large dataset sampled from μ. Thanks to the deveplopment of multi-party and federated learning [Konečný et al., 2016, Zinkevich et al., 2010], this procedure can be done using `SQUERY` with strong privacy guarantee even when the task closure of μ consists of multiple data providers. Furthermore, we can also utilize large-data MRE methods as v_{valid}, in special the direct approaches (See Section 6), which only rely on simple statistics calculated from model predictions. Getting model predictions without fine-tuning is obviously `SQUERY`. This will simplify the validation since no fine-tune process is needed. In the cross-dataset experiments, we use the LEEP score Nguyen et al. [2020] as v_{valid} due to its good performance for large-data MRE under classification.

C More on Experimental Setups

1. Configurations for pre-trained models. When obtaining pre-trained models from MNIST, CIFAR-10, and CIFAR-100, we use ResNet-20 He et al. [2016] as the model structure. When obtaining pre-trained models from dSprites and MiniImageNet, we use ResNet-18 He et al. [2016] as the model structure. All models are trained with 120 epochs over the training datasets using SGD with momentum=0.9. The batch size is 128. The learning rate starts from 0.1, and is divided by 10 on the 60-th and 90-th epochs.

2. Pre-trained model features in synergistic learning. For dSprites, when a model is trained on one domain, we use the three latent factors of that domain as the model feature. For models trained on other datasets, we generate the feature for one of their prediction heads by averaging its predictions over all data for each dataset class to formulate the model feature vectors, such that the feature dimensions are the same to the number of dataset classes.

3. Model structures for the MRE model in synergistic learning. For all experiments, we use the ResNet-12 feature backbone proposed in Oreshkin et al. [2018] as the task feature backbone, and the model feature backbone is a two-layer fully-connected network with 2048-dimensional hidden layers. The output dimensions of the two feature backbones are 640. For the task and model transform modules, all the fully-connected layers have the dimension of 640. We use a transformer-like self-attention structure as the task module, which is motivated by FEAT [Ye et al., 2020].

4. Training configurations for synergistic learning. The MRE model is trained with SGD with momentum=0.9. For feature pre-training, we set the number of iterations to be 10000 and the learn-

Table 2: More Results for the classification experiments. τ and τ_w indicate the Kendall's τ-coefficient and its weighted version calculated from 100 randomly generated five-class target tasks. The results are mean$\pm$95% confidence interval calculated from five random seeds. K indicates the number of training instances per class for each of the testing tasks.

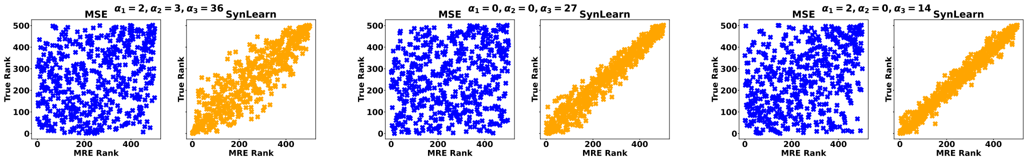
Setting	Method	$K = 5$		$K = 10$		$K = 15$		$K = 20$	
		τ	τ_w	τ	τ_w	τ	τ_w	τ	τ_w
CIFAR-100 → CUB-200-2011	LogME	0.110±0.027	0.270±0.084	0.130±0.010	0.192±0.150	0.160±0.018	0.237±0.117	0.177±0.026	0.378±0.088
	NCE	0.104±0.012	0.189±0.094	0.148±0.024	0.280±0.043	0.176±0.020	0.336±0.026	0.213±0.018	0.380±0.082
	LEEP	0.140±0.017	0.240±0.089	0.189±0.027	0.345±0.054	0.218±0.021	0.390±0.033	0.251±0.020	0.447±0.072
	SynLearn	**0.242±0.013**	**0.421±0.067**	**0.266±0.030**	**0.467±0.056**	**0.269±0.018**	**0.429±0.136**	**0.282±0.022**	**0.484±0.103**
CIFAR-100 → Caltech-256	LogME	0.155±0.073	0.415±0.161	0.227±0.060	0.543±0.159	0.255±0.027	0.586±0.111	0.255±0.021	0.508±0.154
	NCE	0.184±0.022	0.455±0.080	0.270±0.052	0.573±0.067	0.309±0.055	0.631±0.066	0.326±0.066	0.568±0.133
	LEEP	0.245±0.025	0.554±0.090	0.321±0.057	0.629±0.050	0.347±0.059	0.662±0.055	**0.357±0.068**	0.601±0.127
	SynLearn	**0.343±0.035**	**0.637±0.117**	**0.369±0.044**	**0.656±0.119**	**0.359±0.063**	**0.690±0.093**	0.352±0.086	**0.615±0.169**
MiniImageNet → CUB-200-2011	LogME	0.132±0.021	0.182±0.240	0.167±0.038	0.336±0.130	0.189±0.026	0.363±0.232	0.226±0.044	0.442±0.117
	NCE	0.125±0.017	0.160±0.103	0.197±0.024	0.275±0.071	0.241±0.020	0.429±0.061	0.271±0.024	0.422±0.115
	LEEP	0.181±0.020	0.289±0.108	0.261±0.025	0.412±0.103	0.294±0.018	0.521±0.025	**0.316±0.026**	**0.500±0.081**
	SynLearn	**0.295±0.048**	**0.476±0.122**	**0.315±0.040**	**0.529±0.077**	**0.306±0.016**	**0.564±0.076**	**0.316±0.032**	0.494±0.092
MiniImageNet → Caltech-256	LogME	0.146±0.094	0.382±0.137	0.212±0.068	0.430±0.200	0.239±0.032	0.464±0.151	0.247±0.019	0.500±0.110
	NCE	0.163±0.029	0.290±0.066	0.234±0.044	0.401±0.169	0.267±0.040	0.478±0.131	0.280±0.049	0.451±0.089
	LEEP	0.208±0.037	0.350±0.080	0.278±0.049	0.470±0.146	0.303±0.044	0.511±0.128	0.310±0.049	0.500±0.108
	SynLearn	**0.325±0.020**	**0.519±0.140**	**0.349±0.013**	**0.568±0.152**	**0.358±0.020**	**0.590±0.186**	**0.344±0.046**	**0.544±0.159**

ing rate is fixed to 0.001. For metric learning, we set the total number of iterations to be 40000. The initial learning rate is 0.001 and is divided by 10 under the 20000-th iteration. However, if feature pre-training is used, during metric learning, we fix the learning rate for updating the task feature backbone to be 0.0001 in all metric learning iterations. In each iteration, the model batch size is set to be the number of all pre-trained models used for training. The task batch size is set to 5. By defalut, we set $\gamma = 0.5, \gamma_{pre} = 5, \gamma_1 = 0.7, \gamma_2 = 0.2, w_{att} = 1$ and training $K = 10$.

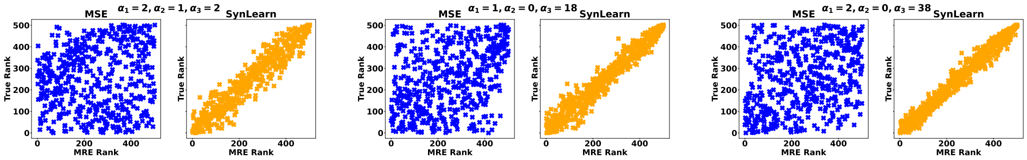
5. Testing setup for the cross-dataset experiments. For each of the testing task, we randomly generate the target training datasets based on the testing K. We further set the number of head re-training epoches to be 100. Except for using CUB-200-2011 and Caltech-256 as the target datasets, under which 10 instances per class are used as the testing data, we set the number of testing instances per class as 50 to test the prediction accuracy of the transferred models.

D Additional Experimental Results

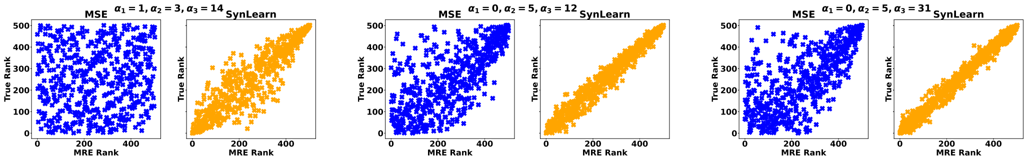
In this section, we provide three additional experimental results. Figure 5 illustrates more results on dSprites additional to Figure 4. Figure 6 illustrates the ablation studies on attention supervision weight w_{att} and the training K in the classification experiments. The ablation studies show that the attention supervision is indeed helpful for learning, meanwhile the performance of synergistic learning is relatively stable under the change of the training K. Furthermore, we provide additional cross-dataset results in Table 2. We use two additional datasets as the target datasets, CUB-200-2011 [Wah et al., 2011] and Caltech-256 [Griffin et al., 2007], to further verify the performance of SynLearn. We still use CIFAR-100 and MiniImageNet as the source datasets to generate the pre-trained models. For both CUB-200-2011 and Caltech-256, we randomly select 30 instances in each class to form the test set and the remaining instances are used for synergistic learning. The other experimental configurations are similar to the experiments in Table 1, except for that we do not include feature pre-training and use the full training data for metric learning. This is due to the fact that the numbers of training instances per class for both datasets are small. Even though it is more challenging for synergistic learning to learn with small-sized training data, from Table 2, we still observe similar performance gain of synergistic learning over the comparison methods.



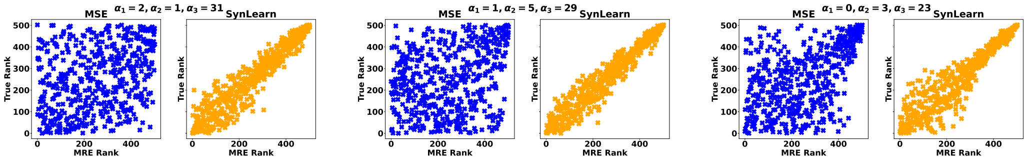
(a) In-distribution domain 4 (b) In-distribution domain 5 (c) In-distribution domain 6



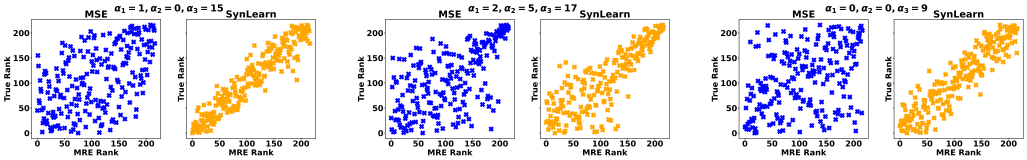
(d) In-distribution domain 7 (e) In-distribution domain 8 (f) In-distribution domain 9



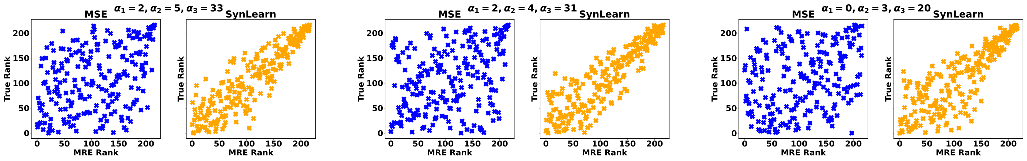
(g) In-distribution domain 10 (h) In-distribution domain 11 (i) In-distribution domain 12



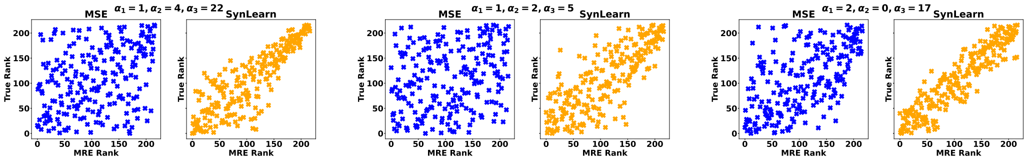
(j) In-distribution domain 13 (k) In-distribution domain 14 (l) In-distribution domain 15



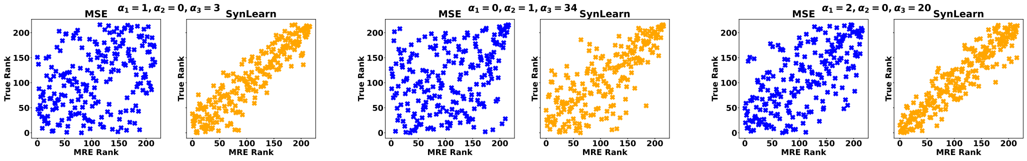
(m) Out-distribution domain 4 (n) Out-distribution domain 5 (o) Out-distribution domain 6



(p) Out-distribution domain 7 (q) Out-distribution domain 8 (r) Out-distribution domain 9

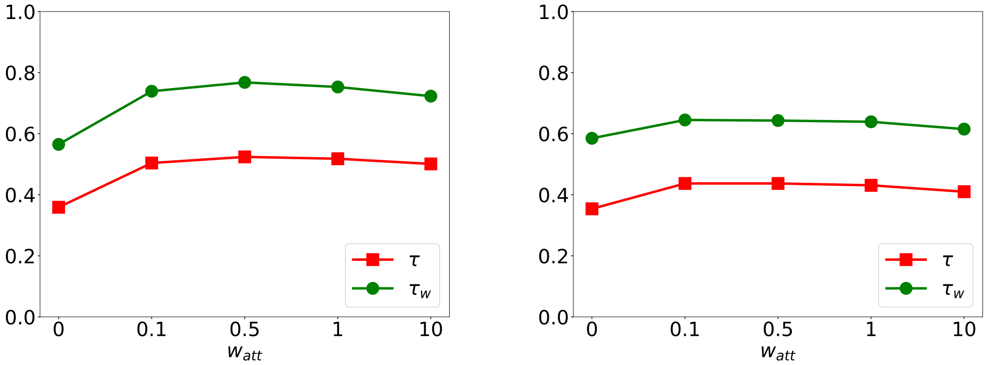


(s) Out-distribution domain 10 (t) Out-distribution domain 11 (u) Out-distribution domain 12

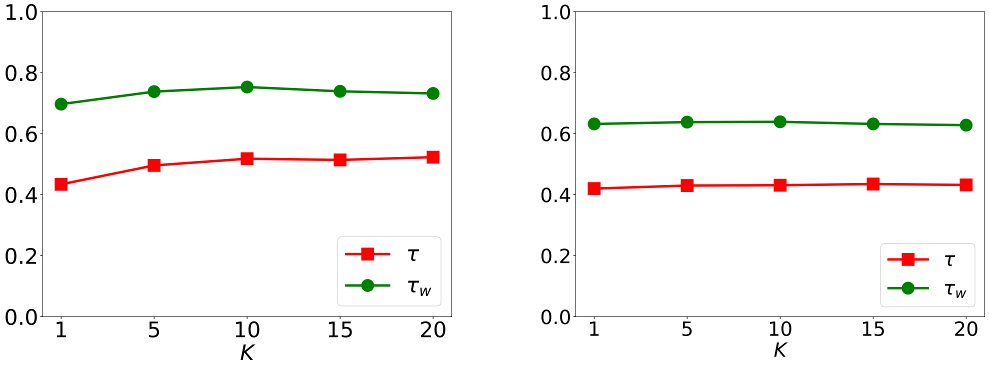


(v) Out-distribution domain 13 (w) Out-distribution domain 14 (x) Out-distribution domain 15

Figure 5: More results on dSprites following Figure 4. (a-l) show in-distribution results and (m-x) show out-distribution results.



(a) Vary w_{att} for CIFAR-100→MiniImageNet (b) Vary w_{att} for MiniImageNet→CIFAR-100



(c) Vary training K for CIFAR-100→MiniImageNet (d) Vary training K for MiniImageNet→CIFAR-100

Figure 6: Ablation studies on w_{att} and training K for the cross-dataset classification experiments. Testing K is fixed to 20 for all results.