Active Multi-Task Representation Learning

Yifang Chen [1]   Simon S. Du [1]   Kevin Jamieson [1]

Abstract

To leverage the power of big data from source tasks and overcome the scarcity of the target task samples, representation learning based on multi-task pretraining has become a standard approach in many applications. However, up until now, choosing which source tasks to include in the multi-task learning has been more art than science. In this paper, we give the first formal study on resource task sampling by leveraging the techniques from active learning. We propose an algorithm that iteratively estimates the relevance of each source task to the target task and samples from each source task based on the estimated relevance. Theoretically, we show that for the linear representation class, to achieve the same error rate, our algorithm can save up to a *number of source tasks* factor in the source task sample complexity, compared with the naive uniform sampling from all source tasks. We also provide experiments on real-world computer vision datasets to illustrate the effectiveness of our proposed method on both linear and convolutional neural network representation classes. We believe our paper serves as an important initial step to bring techniques from active learning to representation learning.

1. Introduction

Much of the success of deep learning is due to its ability to efficiently learn a map from high-dimensional, highly-structured input like natural images into a dense, relatively low-dimensional representation that captures the semantic information of the input. Multi-task learning leverages the observation that similar tasks may share a common representation to train a single representation to overcome a scarcity

---

[*]Equal contribution [1]Paul G. Allen School of Computer Science & Engineering, University of Washington. Correspondence to: Yifang Chen <yifangc@cs.washington.edu>, Simon S. Du <ssdu@cs.washington.edu>, Kevin Jamieson <jamieson@cs.washington.edu>.

of data for any one task. In particular, given only a small amount of data for a target task, but copious amounts of data from source tasks, the source tasks can be used to learn a high-quality low-dimensional representation, and the target task just needs to learn the map from this low-dimensional representation to its target-specific output. This paradigm has been used with great success in natural language processing domains GPT-2 (Radford et al.), GPT-3 (Brown et al., 2020), Bert (Devlin et al., 2018), as well as vision domains CLIP (Radford et al., 2021).

This paper makes the observation that not all tasks are equally helpful for learning a representation, and a priori, it can be unclear which tasks will be best suited to maximize performance on the target task. For example, modern datasets like CIFAR-10, ImageNet, and the CLIP dataset were created using a list of search terms and a variety of different sources like search engines, news websites, and Wikipedia. (Krizhevsky, 2009; Deng et al., 2009; Radford et al., 2021) Even if more data always leads to better performance, practicalities demand some finite limit on the size of the dataset that will be used for training. Up until now, choosing which source tasks to include in multi-task learning has been an ad hoc process and more art than science. *In this paper, we aim to formalize the process of prioritizing source tasks for representation learning by formulating it as an active learning problem.*

Specifically, we aim to achieve a target accuracy on a target task by requesting as little total data from source tasks as possible. For example, if a target task was to generate captions for images in a particular domain where few examples existed, each source task could be represented as a search term into Wikipedia from which (image, caption) pairs are returned. By sampling moderate numbers of (image, caption) pairs resulting from each search term (task), we can determine which tasks result in the best performance on the source task and increase the rate at which examples from those terms are sampled. By quickly identifying which source tasks are useful for the target task and sampling only from those, we can reduce the overall number of examples to train over, potentially saving time and money. Moreover, prioritizing relevant tasks in training, in contrast to uniformly weighting them, even has the potential to improve performance, as demonstrated in (Chen et al., 2021).

From the theoretical perspective, Tripuraneni et al. (2020; 2021); Du et al. (2020) study few-shots learning via multi-task representation learning and gives the generalization guarantees, that, such representation learning can largely reduce the target sample complexity. But all those works only consider *uniform sampling* from each source task and thus establish the proof based on benign diversity assumptions on the sources tasks as well as some common assumptions between target and source tasks.

In this paper, we initiate the systematic study on using active learning to sample from source tasks. We aim to achieve the following two goals:

1. If there is a fixed budget on the source task data to use during training, we would like to select sources that maximize the accuracy of target task relative to naive uniform sampling from all source tasks. Equivalently, to achieve a given error rate, we want to reduce the amount of required source data. In this way, we can reduce the computation because the training complexity generally scales with the amount of data used, especially when the user has limited computing resources (e.g., a finite number of GPUs).

2. Given a target task, we want to output a relevance score for each source task, which can be useful in at least two aspects. First, the scores suggest which certain source tasks are helpful for the target task and inform future task or feature selection (sometimes the task itself can be regard as some latent feature). Second, the scores help the user to decide which tasks to sample more, in order to further improve the target task accuracy.

1.1. Our contributions

In our paper, given a single target task and M source tasks we propose a novel quantity $\nu^* \in \mathbb{R}^M$ that characterizes the relevance of each source task to the target task (cf. Defn 3.1). We design an active learning algorithm which can take any representation function class as input. The algorithm iteratively estimates ν^* and samples data from each source task based on the estimated ν^*. The specific contributions are summarized below:

- In Section 3, we give the definition of ν^*. As a warm up, we prove that when the representation function class is linear and ν^* is known, if we sample data from source tasks according to the given ν^*, the sample complexity of the source tasks scales with the sparsity of $\nu^* \in \mathbb{R}^M$ (the m-th task is relevant if $\nu_m^* \neq 0$). This can save up to a factor of M, the *number of source tasks*, compared with the naive uniform sampling from all source tasks.

- In Section 4, we drop the assumption of knowing ν^*

and describe our active learning algorithm that iteratively samples examples from tasks to estimate ν^* from data. We prove that when the representation function class is linear, our algorithm never performs worse than uniform sampling, and achieves a sample complexity nearly as good as when ν^* is known. The key technical innovation here is to have a trade-off on less related source tasks between saving sample complexity and collecting sufficient informative data for estimating ν^*.

- In Section 5, we empirically demonstrate the effectiveness of our active learning algorithm by testing it on the corrupted MNIST dataset with both linear and convolutional neural network (CNN) representation function classes. The experiments show our algorithm gains substantial improvements compared to the non-adaptive algorithm on both models. Furthermore, we also observe that our algorithm generally outputs higher relevance scores for source tasks that are semantically similar to the target task.

1.2. Related work

There are many existing works on provable *non-adaptive* representation learning with various assumptions. Tripuraneni et al. (2020; 2021); Du et al. (2020); Thekumparampil et al. (2021); Collins et al. (2021); Xu & Tewari (2021) assume there exists an underlying representation shared across all tasks. (Notice that some works focus on learning a representation function for *any possible target task*, instead of learning a model for a specific target task as is the case in our work.) In particular, Tripuraneni et al. (2020); Thekumparampil et al. (2021) assume a low dimension linear representation. Furthermore, it assumes the covariance matrix of all input features is the identity and the linear representation model is orthonormal. Du et al. (2020); Collins et al. (2021) also study a similar setting but lift the identity covariance and orthonormal assumptions. Both works obtain similar conclusions. We will discuss our results in the context of these two settings in Section 2.

Going beyond the linear representation, Du et al. (2020) generalize their bound to a 2-layer ReLu network and Tripuraneni et al. (2021) further considers any general representation and linear predictor classes. More recent work has studied fine-tuning in both theoretical and empirical contexts Shachaf et al. (2021); Chua et al. (2021); Chen et al. (2021). We leave extending our theoretical analysis to more general representation function classes as future work. Other than the generalization perspective, Tripuraneni et al. (2021); Thekumparampil et al. (2021); Collins et al. (2021) propose computational efficient algorithms in solving this non-convex empirical minimization problems during representation learning, including Method-of-moments (MOM) algorithm and Alternating Minimization. Incorpo-

rating these efficient algorithms into our framework would also be a possible direction in the future.

Chen et al. (2021) also consider learning a weighting over tasks. However, their motivations are much different since they are working under the hypothesis that some tasks are not only irrelevant, but even *harmful* to include in the training of a representation. Thus, during training they aim to down-weight potentially harmful source tasks and up-weight those source tasks most relevant to the target task. But the critical difference between their work and ours is that they assume a pass over the complete datasets from all tasks is feasible whereas we assume it is not (e.g., where each task is represented by a search term to Wikipedia or Google). In our paper, their setting would amount to being able to solve for ν^* for free, the equivalent of the "known ν^*" setting of our warm-up section. However, our main contribution is an active learning algorithm that ideally only looks at a vanishing fraction of the data from all the sources to train a representation.

There exists some empirical multi-task representation learning/transfer learning works that have similar motivations as us. For example, Yao et al. (2021) use a heuristic retriever method to select a subset of target-related NLP source tasks and show training on a small subset of source tasks can achieve similar performance as large-scale training. Zamir et al. (2018); Devlin et al. (2018) propose a transfer learning algorithm based on learning the underlying structure among visual tasks, which they called Taskonomy, and gain substantial experimental improvements.

Many classification, regression, and even optimization tasks may fall under the umbrella term *active learning* (Settles, 2009). We use it in this paper to emphasize that a priori, it is unknown which source tasks are relevant to the target task. We overcome this challenge by iterating the closed-loop learning paradigm of 1) collect a small amount of data, 2) make inferences about task relevancy, and 3) leverage these inferences to return to 1) with a more informed strategy for data collection.

2. Preliminaries

In this section, we formally describe our problem setup which will be helpful for our theoretical development.

Problem setup. Suppose we have M source tasks and one target task, which we will denote as task $M + 1$. Each task $m \in [M + 1]$ is associated with a joint distribution μ_m over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^d$ is the input space and $\mathcal{Y} \in \mathbb{R}$ is the output space. We assume there exists an underlying representation function $\phi^* : \mathcal{X} \to \mathcal{Z}$ that maps the input to some feature space $\mathcal{Z} \in \mathbb{R}^K$ where $K \ll d$. We restrict the representation function to be in some function

class Φ, e.g., linear functions, convolutional nets, etc. We also assume the linear predictor to be a linear mapping from feature space to output space, which is represented by $w_m^* \in \mathbb{R}^K$. Specifically, we assume that for each task $m \in [M + 1]$, an i.i.d sample $(x, y) \sim \mu_m$ can be represented as $y = \phi(x)^\top w_m^* + z$, where $z \sim \mathcal{N}(0, \sigma^2)$. Lastly, we also impose a regularity condition such that for all m, the distribution of x when $(x, y) \sim \mu_m$ is 1-sub-Gaussian.

During the learning process, we assume that we have only a small, fixed amount of data $\{x_{M+1}^i, y_{M+1}^i\}_{i \in [n_{M+1}]}$ drawn i.i.d. from the target task distribution μ_{M+1}. On the other hand, at any point during learning we assume we can obtain an i.i.d. sample from any source task $m \in [M]$ without limit. This setting aligns with our main motivation for active representation learning where we usually have a limited sample budget for the target task but nearly unlimited access to large-scale source tasks (such as (image,caption) example pairs returned by a search engine from a task keyword).

Our goal is to use as few total samples from the source tasks as possible to learn a representation and linear predictor ϕ, w_{M+1} that minimizes the excess risk on the target task defined as

$$\text{ER}_{M+1}(\phi, w) = L_{M+1}(\phi, w) - L_{M+1}(\phi^*, w_{M+1}^*)$$

where $L_{M+1}(\phi, w) = \mathbb{E}_{(x,y) \sim \mu_{M+1}} \left[(\langle \phi(x), w \rangle - y)^2 \right]$.

Our theoretical study focuses on the linear representation function class, which is studied in (Du et al., 2020; Tripuraneni et al., 2020; 2021; Thekumparampil et al., 2021).

Definition 2.1 (low-dimension linear representation). $\Phi = \{x \to B^\top x \mid B \in \mathbb{R}^{d \times K}\}$. We denote the true underlying representation function as B^*. Without loss of generality, we assume for all $m \in [M + 1]$, $\mathbb{E}_{\mu_m}[xx^\top]$ are equal.

We also make the following assumption which has been used in (Tripuraneni et al., 2020). We note that Theorem 3.2 does not require this assumption, but Theorem E.4 does.

Assumption 2.2 (Benign low-dimension linear representation). We assume $\mathbb{E}_{\mu_m}[xx^\top] = I$ and $\Omega(1) \leq \|w_m^*\|_2 \leq R$ for all $m \in [M + 1]$. We also assume B^* is not only linear, but also orthonormal.

Notations We denote the n_m i.i.d samples collected from source task m as the input matrix $X_m \in \mathbb{R}^{n_m \times d}$, output vector $Y_m \in \mathbb{R}^{n_m}$ and noise vector $Z_m \in \mathbb{R}^{n_m}$. We then denote the expected and empirical input variances as $\Sigma_m = \mathbb{E}_{(x,y) \sim \mu_m} xx^\top$ and $\hat{\Sigma}_m = \frac{1}{n_m}(X_m)^\top X_m$. In addition, we denote the collection of $\{w_m\}_{m \in [M]}$ as $W \in \mathbb{R}^{K \times M}$. Note that, the learning process will be divided into several epochs in our algorithm stated later, so we sometimes add subscript or superscript i on those empirical notations to refer to the data used in certain epoch i. Finally, we use $\tilde{\mathcal{O}}$ to hide $\log(K, M, d, 1/\varepsilon, \sum_{m=1}^M n_m)$.

Other data assumptions Based on our large-scale source tasks motivation, we assume $M \geq K$ and $\sigma_{\min}(W^*) > 0$, which means the source tasks are diversified enough to learn all relevant representation features with respect to the low-dimension space. This is the standard diversity assumption used in many recent works (Du et al., 2020; Tripuraneni et al., 2020; 2021; Thekumparampil et al., 2021). In addition, we assume $\sigma \geq \Omega(1)$ to make our main result easier to read. This assumption can be lift by adding some corner case analysis.

3. Task Relevance ν^* and More Efficient Sampling with Known $\nu*$

In this section, we give our key definition of task relevance, based on which, we design a more efficient source task sampling strategy.

Note because $\sigma_{\min}(W^*) > 0$, we can regard w_{M+1}^* as a linear combination of $\{w_m^*\}_{m \in [M]}$.

Definition 3.1. $\nu^* \in \mathbb{R}^M$ is defined as

$$\nu^* = \arg\min_{\nu} \|\nu\|_2 \qquad \text{s.t.} \quad W^*\nu = w_{M+1}^* \quad (1)$$

where larger $|\nu^*(m)|$ means higher relevance between source task m and the target task. If ν^* is known to the learner, intuitively, it makes sense to draw more samples from source tasks that are most relevant.

For each source task $m \in [M]$, Line 3 in Alg. 1 draws $n_m \propto (\nu^*(m))^2$ samples. The algorithm then estimates the shared representation $\phi : \mathbb{R}^d \to \mathbb{R}^K$, and task-specific linear predictors $W = \{w_m^*\}_{m=1}^M$ by empirical risk minimization across all source tasks following the standard multi-task representation learning approach.

Below, we give our theoretical guarantee on the sample complexity from the source tasks when ν^* is known.

Theorem 3.2. *Under the low-dimension linear representation setting as defined in Definition 2.1, with probability at least $1 - \delta$, our algorithm's output satisfies $\mathrm{ER}(\hat{B}, \hat{w}_{M+1}) \leq \varepsilon^2$ whenever the total sampling budget from all sources N_{total} is at least*

$$\widetilde{\mathcal{O}}\left((Kd + KM + \log(1/\delta))\sigma^2 s^* \|\nu^*\|_2^2 \varepsilon^{-2}\right)$$

and the number of target samples n_{M+1} is at least

$$\widetilde{\mathcal{O}}(\sigma^2(K + \log(1/\delta))\varepsilon^{-2})$$

where $s^* = \min_{\gamma \in [0,1]}(1-\gamma)\|\nu^*\|_{0,\gamma} + \gamma M$ and $\|\nu\|_{0,\gamma} := \left|\left\{m : |\nu_m| > \sqrt{\gamma \frac{\|\nu^*\|_2^2}{N_{total}}}\right\}\right|$.

Note that the number of target samples n_{M+1} scales only with the dimension of the feature space K, and *not* the input

---

Algorithm 1 Multi-task sampling strategy with Known ν^*

1: **Input:** confidence δ, representation function class Φ, combinatorial coefficient ν^*, source-task sampling budget $N_{\text{total}} \gg M(Kd + \log(M/\delta))$

2: Initialize the lower bound $\underline{N} = Kd + \log(M/\delta)$ and number of samples $n_m = \max\left\{(N_{\text{total}} - M\underline{N})\frac{(\nu^*(m))^2}{\|\nu^*\|_2^2}, \underline{N}\right\}$ for all $m \in [M]$.

3: For each task m, draw n_m i.i.d samples from the corresponding offline dataset denoted as $\{X_m, Y_m\}_{m=1}^M$

4: Estimate the models as

$$\hat{\phi}, \hat{W} = \arg\min_{\phi \in \Phi, W=[w_1,\ldots,w_M]} \sum_{m=1}^M \|\phi(X_m)w_m - Y_m\|^2.$$
(2)

$$\hat{w}_{M+1} = \arg\min_w \|\hat{\phi}(X_{M+1})w - Y_{M+1}\|^2 \quad (3)$$

5: Return $\hat{\phi}, \hat{w}_{M+1}$

---

dimension $d \gg K$ which would be necessary without multi-task learning. This dependence is known to be optimal (Du et al., 2020). The quantity s^* characterizes our algorithm's ability to adapt to the approximate sparsity of ν^*. Noting that $\sqrt{\frac{\|\nu^*\|_2^2}{N_{total}}}$ is roughly on the order of ε, taking $\gamma \approx 1/M$ suggests that to satisfy $\mathrm{ER}(\hat{B}, \hat{w}_{M+1}) \leq \varepsilon^2$, only those source tasks with relevance $|\nu^*(m)| \gtrsim \varepsilon$ are important for learning.

For comparison, we rewrite the bound in (Du et al., 2020) in the form of ν^*.

Theorem 3.3. *Under Assumption 2.1, to obtain the same accuracy result, the non-adaptive (uniform) sampling of (Du et al., 2020) requires that the total sampling budget from all sources N_{total} is at least*

$$\widetilde{\mathcal{O}}\left((Kd + KM + \log(1/\delta))\sigma^2 M\|\nu^*\|_2^2\varepsilon^{-2}\right)$$

and requires the same amount of target samples as above.

Note the key difference is that the s^* in Theorem 3.2 is replaced by M in Theorem 3.3. Below we give a concrete example to show this difference is significant.

Example: Sparse ν^*. Consider an extreme case where $w_m = e_{m \bmod (K-1)+1}$ for all $m \in [M-1]$, and $w_M = w_{M+1} = e_K$. This suggests that the target task is exactly the same as the source task M and all the other source tasks are uninformative. It follows that ν^* is a 1-sparse vector e_M and $s^* = 1$ when $\gamma = 0$. We conclude that uniform sampling requires a sample complexity that is M times larger than that of our non-uniform procedure.

3.1. Proof sketch of Theorem 3.2

We first claim two inequalities that are derived via straight-forward modifications of the proofs in Du et al. (2020):

$$\text{ER}(\hat{B}, \hat{w}_{M+1}) \lesssim \frac{\|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^* w^*_{M+1}\|^2}{n_{M+1}} \quad (4)$$

$$\frac{\|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^* \widetilde{W}^*\|^2_F}{n_{M+1}} \lesssim \sigma^2 \left(K(M+d) + \log\frac{1}{\delta}\right) \quad (5)$$

where $P^{\perp}_A = I - A\left(A^\top A\right)^\dagger A^\top$, $\tilde{\nu}^*(m) = \frac{\nu^*(m)}{\sqrt{n_m}}$, and $\tilde{W}$ is $\left[\sqrt{n_1}w^*_1, \sqrt{n_2}w^*_2, \ldots, \sqrt{n_M}w^*_M\right]$. By using these two results and noting that $w^*_{M+1} = \widetilde{W}^*\tilde{\nu}^*$, we have

$$\text{ER}(\hat{B}, \hat{w}_{M+1}) \overset{(4)}{\lesssim} \frac{1}{n_{M+1}}\|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^*\widetilde{W}^*\tilde{\nu}^*\|^2_2$$

$$\leq \frac{1}{n_{M+1}}\|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^*\widetilde{W}^*\|^2_F \|\tilde{\nu}^*\|^2_2$$

$$= (5) \times \|\tilde{\nu}^*\|^2_2.$$

The key step to our analysis is the decomposition of $\|\tilde{\nu}^*\|^2_2$. If we denote $\epsilon^{-2} = \frac{N_{\text{total}}}{\|\nu^*\|^2_2}$, we have, for any $\gamma \in [0,1]$,

$$\sum_m \frac{\nu^*(m)^2}{n_m}\left(\mathbf{1}\{|\nu^*(m)| > \sqrt{\gamma}\epsilon\} + \mathbf{1}\{|\nu^*(m)| \leq \sqrt{\gamma}\epsilon\}\right)$$

$$\lesssim \sum_m \left(\epsilon^2\mathbf{1}\{|\nu^*(m)| > \sqrt{\gamma}\epsilon\} + \gamma\epsilon^2\mathbf{1}\{|\nu^*(m)| \leq \sqrt{\gamma}\epsilon\}\right)$$

where the inequality comes from the definition of n_m and the fact $N_{\text{total}} \gg M\underline{N}$. Now by replacing the value of ϵ and $\|\nu\|_{0,\gamma}$, we get the desired result.

4. Main Algorithm and Theory

In the previous section, we showed the advantage of target-aware source task sampling when the optimal mixing vector ν^* between source tasks and the target task is *known*. In practice, however, ν^* is unknown and needs to be estimated based on the estimation of W^* and w^*_{M+1}, which are themselves consequences of the unknown representation ϕ^*. In this section, we design an algorithm that adaptively samples from source tasks to efficiently learn ν^* and the prediction function for the target task $B^*w^*_{M+1}$. The pseudocode for the procedure is found in Alg. 2.

We divide the algorithm into several epochs. At the end of each epoch i, we obtain estimates $\hat{\phi}_i$, $\hat{W}_i$ and $\hat{w}^i_{M+1}$ which are then used to calculate the task relevance denoted as $\hat{\nu}_{i+1}$. Then in the next epoch $i+1$, we sample data based on $\hat{\nu}_{i+1}$. The key challenge in this iterative estimation approach is that the error of the estimation propagates from round to round due to unknown ν^* if we directly apply the sampling strategy proposed in Section 3. To avoid inconsistent estimation, we enforce the condition that each source task is sampled at least $\beta\epsilon_i^{-1}$ times to guarantee that $|\hat{\nu}_i(m)|$ is

---

Algorithm 2 `Active Task Relevance Sampling`

1: **Input:** confidence δ, a lower bound of $\sigma_{\min}(W^*)$ as $\underline{\sigma}$, representation function class Φ
2: Initialize $\hat{\nu}_1 = [1/M, 1/M, \ldots]$, $\epsilon_i = 2^{-i}$ and $\{\beta_i\}_{i=1,2,\ldots}$, which will be specified later
3: **for** $i = 1, 2, \ldots$ **do**
4:   Set $n^i_m = \max\left\{\beta_i\hat{\nu}_i^2(m)\epsilon_i^{-2}, \beta_i\epsilon_i^{-1}\right\}$.
5:   For each task m, draw n_m i.i.d samples from the corresponding offline dataset denoted as $\{X^i_m, Y^i_m\}^M_{m=1}$
6:   Estimate $\hat{\phi}^i, \hat{W}_i, \hat{w}^i_{M+1}$ with Eqn. (2) and (3)
7:   Estimate the coefficient as
   $$\hat{\nu}_{i+1} = \arg\min_\nu \|\nu\|^2_2 \quad \text{s.t.} \quad \hat{W}_i\nu = \hat{w}^i_{M+1} \quad (6)$$
8: **end for**

---

always $\sqrt{\epsilon_i}$-close to $|c\nu^*(m)|$, where $c \in [1/16, 4]$. We will show why such estimation is enough in our analysis.

4.1. Theoretical results under linear representation

Here we give a theoretical guarantee for the realizable linear representation function class. Under this setting, we choose β as follows [1]

$$\beta := \beta_i = \left(3000K^2R^2(KM + Kd\log(\frac{N_{\text{total}}}{\varepsilon M}))\right.$$
$$\left. + \log(\frac{M\log(1/N_{\text{total}})}{\delta/10})\right)\frac{1}{\underline{\sigma}^6}, \forall i$$

Theorem 4.1. *Suppose we know in advance a lower bound of $\sigma_{\min}(W^*)$ denoted as $\underline{\sigma}$. Under the benign low-dimension linear representation setting as defined in Assumption 2.2, we have $\text{ER}(\hat{B}, \hat{w}_{M+1}) \leq \varepsilon^2$ with probability at least $1 - \delta$ whenever the number of source samples N_{total} is at least*

$$\widetilde{\mathcal{O}}\left(\left(K(M+d) + \log\frac{1}{\delta}\right)\sigma^2 s^*\|\nu^*\|^2_2\varepsilon^{-2} + \Box\underline{\sigma}\varepsilon^{-1}\right)$$

where $\Box = \left(MK^2dR/\underline{\sigma}^3\right)\sqrt{s^*}$ and the target task sample complexity n_{M+1} is at least

$$\widetilde{\mathcal{O}}\left(\sigma^2K\varepsilon^{-2} + \Diamond\sqrt{s^*}\underline{\sigma}\varepsilon^{-1}\right)$$

where $\Diamond = \min\left\{\frac{\sqrt{R}}{\underline{\sigma}^2K}, \sqrt{K(M+d) + \log\frac{1}{\delta}}\right\}$ and s^* has been defined in Theorem 3.2.

Discussion. Comparing to the known ν^* case studied in the previous section, in this unknown ν^* setting our algorithm only requires an additional low order term $\Box\underline{\sigma}\varepsilon^{-1}$ to

---

[1]The choice of β here is purely for theoretical proof under the restricted realizable linear representation functions. In practice, we find that the choice of β can be more flexible as long as the lower bound of number of samples is in proportional to ϵ_i^{-1}.

achieve the same objective (under the additional assumption of Assumption 2.2). Also, as long as $\Diamond \leq \widetilde{\mathcal{O}}(\sigma K \varepsilon^{-1})$, our target task sample complexity $\widetilde{\mathcal{O}}(\sigma^2 K \varepsilon^{-2})$ remains the optimal rate (Du et al., 2020).

Finally, we remark that a limitation of our algorithm is that it requires some prior knowledge of $\underline{\sigma}$. However, because it only hits the low-order ϵ^{-1} terms, this is unlikely to dominate either of the sample complexities for reasonable values of d, K, and M.

4.2. Proof sketch

Step 1: We first show that the estimated distribution over tasks $\hat{\nu}_i$ is close to the underlying ν^*.

Lemma 4.2 (Closeness between $\hat{\nu}_i$ and ν^*). *With probability at least $1 - \delta$, for any i, as long as $n_{M+1} \geq \frac{2000\epsilon_i^{-1}}{\underline{\sigma}^4}$, we have*

$$|\hat{\nu}_{i+1}(m)| \in \begin{cases} [|\nu^*(m)|/16, 4|\nu^*(m)|] & \text{if } \nu^*(m) \geq \sigma\sqrt{\epsilon_i} \\ [0, 4\sqrt{\epsilon_i}] & \text{if } |\nu^*(m)| \leq \sigma\sqrt{\epsilon_i} \end{cases}$$

Notice that the sample lower bound in the algorithm immediately implies sufficiently good estimation in the next epoch even if $\hat{\nu}_{i+1}$ goes to 0.

Proof sketch:

Under Assumption 2.2, by solving Eqn (3), we can rewrite the optimization problem on ν^* and $\hat{\nu}_i$ defined in Eqn.(1) and (6) roughly as the follows (see the formal definition in the proof of Lemma E.1 in Appendix E)

$$\hat{\nu}_{i+1} = \arg\min_{\nu} \|\nu\|_2^2$$

$$\text{s.t.} \quad \sum_m \hat{B}_i^\top \left(B^* w_m^* + \frac{1}{n_m^i} \left(X_m^i \right)^\top Z_m \right) \nu(m)$$

$$= \hat{B}_i^\top \left(B^* w_{M+1}^* + \frac{1}{n_{M+1}^i} \left(X_{M+1}^i \right)^\top Z_{M+1} \right),$$

and $\nu^* = \arg\min_{\nu} \|\nu\|_2^2$

$$\text{s.t.} \quad \sum_m w_m^* \nu(m) = w_{M+1}^*.$$

Solving these two optimization problem gives,

$$\nu^*(m) = (B^* w_m^*)^T \left(B^* W^* (B^* W^*)^T \right)^+ (B^* w_{M+1}^*)$$

$$|\hat{\nu}_{i+1}(m)| \leq 2 \left| (B^* w_m^*)^T (\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^T)^+ B^* w_{M+1}^* \right|$$

$$+ \text{ low order noise.}$$

It is easy to see that the main different between these two expressions is $\left(B^* W^* (B^* W^*)^T \right)^+$ and its corresponding empirical estimation. Therefore, by denoting the difference between these two terms as

$$\Delta = (\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^T)^+ - \left(B^* W^* (B^* W^*)^T \right)^+,$$

we can establish the connection between the true and empirical task relevance as

$$|\hat{\nu}_{i+1}(m)| - 2|\nu^*(m)| \lesssim 2 \left| (B^* w_m^*)^T \Delta (B^* w_{M+1}^*) \right| \quad (7)$$

Now the minimization on source tasks shown in Eqn. (2) ensures that

$$\|B^* W^* - \hat{B}_i \hat{W}_i\|_F \leq \sigma \text{poly}(d, M) \sqrt{\epsilon_i}.$$

This helps us to further bound the $(\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^T) - \left(B^* W^* (B^* W^*)^T \right)$ term, which can be regarded as a perturbation on the underlying matrix $B^* W^* (B^* W^*)^T$. Then by using the generalized inverse matrix theorem (Kovanic, 1979), we can show that the inverse of the perturbed matrix is close to its original matrix on some low dimension space.

Therefore, we can upper bound Eqn. (7) by $\sigma\sqrt{\epsilon_i}$. We repeat the same procedure to lower bound the $\frac{1}{2}|\nu^*(m)| - |\hat{\nu}_{i+1}(m)|$. Combining these two, we have

$$|\hat{\nu}_{i+1}(m)| \in \left[\frac{1}{2}|\nu^*(m)| - \frac{7}{16}\sigma\sqrt{\epsilon_i}, 2|\nu^*(m)| + 2\sigma\sqrt{\epsilon_i} \right]$$

This directly lead to the result based on whether $\nu^*(m) \geq \sigma\sqrt{\epsilon_i}$ or not. $\qquad\square$

Step 2: Now we prove the following two main lemmas on the final accuracy and the total sample complexity.

Define event $\mathcal{E}$ as the case that, for all epochs, the closeness between $\hat{\nu}_i$ and ν^* defined Lemma 4.2 has been satisfied.

Lemma 4.3 (Accuracy on each epoch (informal)). *Under $\mathcal{E}$, after the epoch i, we have $\text{ER}(\hat{B}, \hat{w}_{M+1})$ roughly upper bounded by*

$$\frac{\sigma^2}{\beta} \left(KM + Kd + \log\frac{1}{\delta} \right) s_i^* \epsilon_i^2 + \frac{\sigma^2 (K + \log(1/\delta))}{n_{M+1}}$$

where $s_i^* = \min_{\gamma \in [0,1]}(1 - \gamma)\|\nu^*\|_{0,\gamma}^i + \gamma M$ and $\|\nu\|_{0,\gamma}^i := |\{m : \nu_m > \sqrt{\gamma\epsilon_i}\}|$.

Proof sketch: As we showed in Section 3, the key for calculating the accuracy is to upper bound $\sum_m \frac{\nu^*(m)^2}{n_m^i}$. Similarly to Section 3, we employ the decomposition

$$\sum_m \frac{\nu^*(m)^2}{n_m^i} \left(\mathbf{1}\{|\nu^*(m)| > \sqrt{\gamma\epsilon_i}\} + \mathbf{1}\{|\nu^*(m)| \leq \sqrt{\gamma\epsilon_i}\} \right).$$

The last sparsity-related term can again be easily upper bounded by $\mathcal{O}((1 - \|\nu^*\|_{0,\gamma}^i)\sigma^2\gamma\epsilon_i^2)$.

Then in order to make a connection between n_m^i and $\nu^*(m)$ by using Lemma 4.2, we further decompose the first term

as follows and get the upper bound

$$\sum_m \frac{\nu^*(m)^2}{n_m^i} \mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\}$$

$$+ \sum_m \frac{\nu^*(m)^2}{n_m^i} \mathbf{1}\{\sigma\sqrt{\gamma}\epsilon_i \leq |\nu^*(m)| \leq \sqrt{\epsilon_{i-1}}\}$$

$$\lessapprox \sum_m \frac{\hat{\nu}_i^2}{n_m^i} \mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\}$$

$$+ \sum_m \frac{\sigma^2 \epsilon_i}{n_m^i} \mathbf{1}\{\sqrt{\gamma}\epsilon_{i-1} \leq |\nu^*(m)| \leq \sigma\sqrt{\epsilon_{i-1}}\}$$

$$\leq \sum_m \epsilon_i^2/\beta \mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\}$$

$$+ \sum_m \sigma^2 \epsilon_i^2/\beta \mathbf{1}\{\sqrt{\gamma}\epsilon_i \leq |\nu^*(m)| \leq \sigma\sqrt{\epsilon_{i-1}}\}$$

$$\leq \mathcal{O}(\|\nu^*\|_{0,\gamma}^i \epsilon_i^2/\beta)$$

where the second inequality is from the definition of n_m^i.
□

Lemma 4.4 (Sample complexity on each epoch(informal))**.** *Under $\mathcal{E}$, after the epoch i, We have the total number of training samples from source tasks upper bounded by*

$$\mathcal{O}\left(\beta(M\varepsilon^{-1} + \|\nu^*\|_2^2 \varepsilon^{-2})\right) + \text{low-order term} \times \Gamma.$$

Proof sketch: For any fixed epoch i, by definition of n_m^i, we again decompose the summed source tasks based on $\nu^*(m)$ and get the total sample complexity as follows

$$\sum_{m+1}^M \beta\hat{\nu}_i^2(m)\epsilon_i^{-2} \mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\}$$

$$+ \sum_{m+1}^M \beta\hat{\nu}_i^2(m)\epsilon_i^{-2} \mathbf{1}\{|\nu^*(m)| \leq \sigma\sqrt{\epsilon_{i-1}}\} + M\beta\epsilon_i^{-1}$$

Again by replacing the value of $\hat{\nu}$ from Lemma 4.2, we can upper bounded second term in terms of ν^* and we can also show that the third term is low order ϵ^{-1}.
□

Theorem E.4 follows by combining the two lemmas.

5. Experiments

In this section, we empirically evaluate our active learning algorithm for multi-task by deriving tasks from the corrupted MNIST dataset (MNIST-C) proposed in Mu & Gilmer (2019). While our theoretical results only hold for the linear representations, our experiments demonstrate the effectiveness our algorithm on neural network representations as well. We show that our proposed algorithm: (1) achieves better performance when using the same amount of source samples as the non-adaptive sampling algorithm, and (2) gradually draws more samples on important source tasks.

5.1. Experiment setup

Dataset and problem setting. The MNIST-C dataset is a comprehensive suite of 16 different types of corruptions applied to the MNIST test set. To create source and target tasks, we divide each sub-dataset with a specific corruption into 10 tasks by applying one-hot encoding to $0 - 9$ labels. Therefore, we have 160 tasks in total, which we denote as "corruption type + label". For example, *brightness_0* denotes the data corrupted by brightness noise and are relabeled to $1/0$ based on whether the data is number 0 or not. We choose a small number of fixed samples from the target task to mimic the scarcity of target task data. On the other hand, we set no budget limitation on source tasks. We compare the performance of our algorithm to the non-adaptive uniform sampling algorithm, where each is given the same number of source samples and same target task dataset.

Models. We start with the linear representation as defined in our theorem and set $B \in \mathbb{R}^{28*28 \times 50}$ and $w_m^i \in \mathbb{R}^{50}$. Note that although the MNIST problem is usually a classification problem with cross-entropy loss, here we model it as a regression problem with ℓ_2 loss to align with the setting studied in this paper. Moreover, we also test our algorithm with 2-layer ReLU *convolutional neural nets* (CNN) followed by fully-connected linear layers, where all the source tasks share the same model except the last linear layer, also denoted as $w_m^i \in \mathbb{R}^{50}$.

AL algorithm implementation. We run our algorithm iteratively for 4 epochs. The non-adaptive uniform sampling algorithm is provided with the same amount of source samples. There are some difference between our proposed algorithm and what is implemented here. First we re-scale some parameters from the theorem to account for potential looseness in our analysis. Moreover, instead of drawing fresh i.i.d samples for each epoch and discarding the past, in practice, we reuse the samples from previous epochs and only draw what is necessary to meet the required number of samples of the current epoch. This introduces some randomness in the total source sample usage. For example, we may only require 100 samples from the source task A for the current epoch, but we may have sampled 200 from source task A in the previous epoch. So we always sample equal or less than non-adaptive algorithm in a single epoch. Therefore, in our result shown below, the total source sample numbers varies across target tasks. But we argue that this variation are roughly at the same level and will not effect our conclusion. Please refer to Appendix F.1 for details.

5.2. Results

Linear representation. We choose 500 target samples from each target task. After 4 epochs, we use in total around 30000 to 40000 source samples. As a result, our adaptive algorithm frequently outperforms the non-adaptive one as shown in Figure 1.
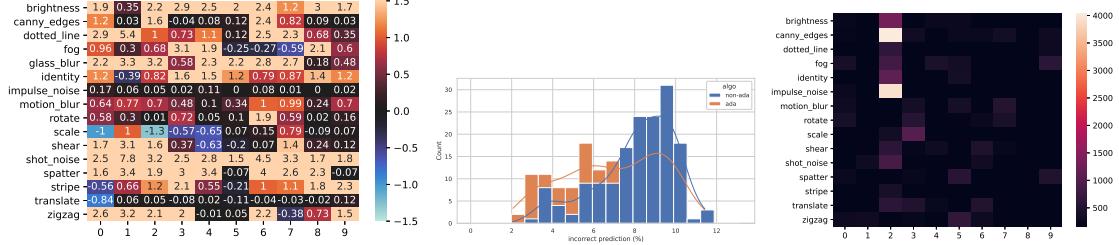
Figure 1. **Performance between the adaptive (ada) and the non-adaptive (non-ada) algorithm on linear representation. Left**: The prediction difference (in %) between ada and non-ada for all target tasks. The larger is the better. Respectively y-axis denotes noise type and x-axis denotes binarized label, with each grid representing a target task, *e.g.*, the grid at the top left corner stands for target task *brightness_0*. In summary, the adaptive algorithm achieves 1.1% higher average accuracy than the non-adaptive one and results same or better accuracy in 136 out of 160 tasks. **Middle**: Histogram summary of incorrect prediction (left is better). There is a clear shift for adaptive algorithm towards left. **Right:** Sampling distribution for the target task *glass_blur_2*. Respectively, the plot shows numbers of samples from each source tasks at the beginning of epoch 3 by running adaptive algorithm. The samples clearly concentrated on several *X_2* source tasks, which meets our intuition that all "2 vs. others" tasks should has closer connection with the *glass_blur_2* target task.
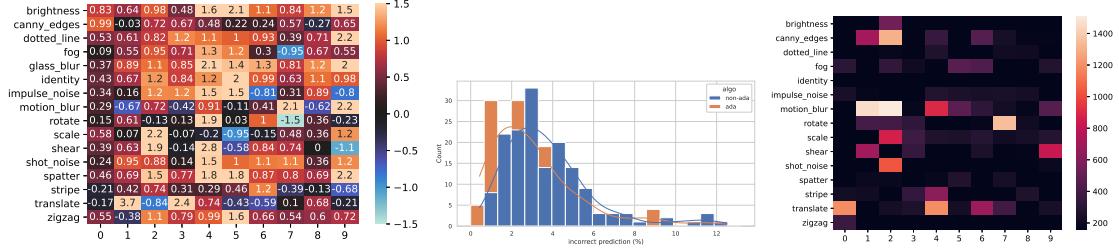

Figure 2. **Performance between the adaptive (ada) and the non-adaptive (non-ada) algorithm on Convnet. Left**: The prediction difference (%) between ada and non-ada for all target tasks. (See more explanations for notations in Figure 1.) In summary, the adaptive algorithm achieves 0.68% higher average accuracy than the non-adaptive one and results same or better accuracy in 133 out of 160 tasks. **Middle**: Histogram summary of incorrect prediction (left is better). There is a clear for adaptive algorithm towards left. Although the average performance improvement is smaller than in the linear representation, the relative improvement is also significant given the already good baseline performance (most prediction error are below 6% while in linear most are above 6%). **Right:** Sample distribution for target task as *glass_blur_2*. A large portion of samples again concentrate on several *X_2* source tasks, which meets our intuition that all "2 vs. others" tasks should has closer connection with *glass_blur_2* target task. But the overall sample distribution is more spread compared to the one on linear representation.

For those cases where gains are not observed, we conjecture that those tasks violate our realizable assumptions more than the others. We provide a more detailed discussion and supporting results for those failure cases in Appendix F.2.2. Next, we investigate the sample distribution at the beginning epoch 3. We show the result for *glass_blur_2* as a representative case with more examples in the Appendix F.3. From the figure, we can clearly see the samples concentrate on more target-related source tasks.

Convnet. We choose 200 target samples from each target task. After 4 epochs, we use in total around 30000 to 40000 source samples. As a result, our adaptive algorithm again frequently outperforms the non-adaptive one as shown in Figure 1. Next, we again investigate the sample distribution at the beginning epoch 3 and show a representative result (more examples in Appendix F.3.2). First of all, there are still a number of source samples again concentrating on "2 vs. others". The reader may notice some other source tasks also contribute a relatively large amount of sample complexity. This is actually a typical phenomenon in our experiment on convnets, which is seldom observed in the

linear representation. This might be due to the more expressive power of CNN that captures some non-intuitive relationships between some source tasks and the target task. Or it might be simply due to the estimation error since our algorithm is theoretically justified only for the realizable linear representation. We provide more discussion in Appendix F.3.1.

6. Conclusion and future work

Our paper takes an important initial step to bring techniques from active learning to representation learning. There are many future directions. From the theoretical perspective, it is natural to analyze some fine-tuned models or even more general models like neural nets as we mentioned in the related work section. From the empirical perspective, our next step is to modify and apply the algorithm on more complicated CV or NLP datasets and further analyze its performance. Finally, it is also interesting to combine our task-wise active learning with instance-wise active learning results.

Acknowledgements

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Chen, S., Crammer, K., He, H., Roth, D., and Su, W. J. Weighted training for cross-task learning, 2021.

Chua, K., Lei, Q., and Lee, J. D. How fine-tuning allows for effective meta-learning, 2021.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. *arXiv preprint arXiv:2102.07078*, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. Few-shot learning via learning the representation, provably. In *International Conference on Learning Representations*, 2020.

Kovanic, P. On the pseudoinverse of a sum of symmetric matrices with applications to estimation. *Kybernetika*, 15(5):(341)–348, 1979. URL http://eudml.org/doc/28097.

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

Mu, N. and Gilmer, J. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Settles, B. Active learning literature survey. 2009.

Shachaf, G., Brutzkus, A., and Globerson, A. A theoretical analysis of fine-tuning with linear teachers. *Advances in Neural Information Processing Systems*, 34, 2021.

Thekumparampil, K. K., Jain, P., Netrapalli, P., and Oh, S. Sample efficient linear meta-learning by alternating minimization. *arXiv preprint arXiv:2105.08306*, 2021.

Tripuraneni, N., Jordan, M., and Jin, C. On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33, 2020.

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

Xu, Z. and Tewari, A. Representation learning beyond linear prediction functions. *arXiv preprint arXiv:2105.14989*, 2021.

Yao, X., Zheng, Y., Yang, X., and Yang, Z. Nlp from scratch without large-scale pretraining: A simple and efficient framework, 2021.

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

A. Appendix structure

In Appendix B, we define the commonly used notations in the following analysis. In Appendix C.2, we define three high probability events and prove three claims as variants of original results in Du et al. (2020). All these events and claims are widely used in the following theoretical analysis. Then we give formal proofs of Theorem 3.2 and Theorem 3.3 in Appendix D and formal proofs of Theorem E.4 in Appendix E. Finally, we show more comprehensive results of experiment in Appendix F.

B. Notation

- Define $\hat{\Delta} = \hat{B}\hat{W} - B^*W^*$ and correspondingly define $\hat{\Delta}^i = \hat{B}_i\hat{W}_i - B^*W^*$ if the algorithm is divided into epochs.

- Define $\hat{\Delta}_m = \hat{B}\hat{w}_m - B^*w_m^*$ and correspondingly define $\hat{\Delta}_m^i = \hat{B}_i\hat{w}_m^i - B^*w_m^*$.

- Restate $P_A^\perp = I - A\left(A^\top A\right)^\dagger A^\top$

- Restate that $\hat{\Sigma}_m = (X_m)^\top X_m$ and correspondingly $\hat{\Sigma}_m^i = (X_m^i)^\top X_m^i$ if the algorithm is divided into epochs.

- Restate that $\Sigma_m = \mathbb{E}\left[\hat{\Sigma}_m\right]$ and correspondingly $\Sigma_m^i = \mathbb{E}\left[\hat{\Sigma}_m^i\right]$.

- Define $\kappa = \frac{\lambda_{max}(\Sigma)}{\lambda_{min}(\Sigma)}$, recall we assume all $\Sigma_m = \Sigma$. Note that in the analysis for adaptive algorithm, we assume identity covariance so $\kappa = 1$.

- For convenience, we write $\sum_{m=1}^M$ as $\sum_m$.

- If the algorithm is divided into epochs, we denote the total number of epoch as Γ.

C. Commonly used claims and definitions

C.1. Co-variance concentration guarantees

We define the following guarantees on the feature covariance concentration that has been used in all proofs below.

$$\mathcal{E}_{\text{source}} = \{0.9\Sigma_m \leq \hat{\Sigma}_m \leq 1.1\Sigma_m, \forall m \in [M]\}$$
$$\mathcal{E}_{\text{target1}} = \{0.9B_1^\top B_2 \leq B_1^\top \hat{\Sigma}_{M+1} B_2 \leq 1.1B_1^\top B_2, \text{for any orthonormal} B_1, B_2 \in \mathbb{R}^{d \times K} \mid \Sigma_{M+1} = I\}$$
$$\mathcal{E}_{\text{target2}} = \{0.9\Sigma_{M+1} \leq B^\top \hat{\Sigma}_{M+1} B \leq 1.1\Sigma_{M+1}, \text{for any } B \in \mathbb{R}^{d \times K}\}$$

By Claim A.1 in Du et al. (2020), we know that, as long as $n_m \gg d + \log(M/\delta), \forall m \in [M+1]$

$$\text{Prob}\left(\mathcal{E}_{\text{source}}^i\right) \geq 1 - \frac{\delta}{10},$$

Moreover, as long as $n_m \gg K + \log(1/\delta)$,

$$\text{Prob}(\mathcal{E}_{\text{target1}}) \geq 1 - \frac{\delta}{10} \qquad\qquad \text{Prob}(\mathcal{E}_{\text{target2}}) \geq 1 - \frac{\delta}{10}.$$

Correspondingly, if the algorithm is divided into epochs where each epoch we draw new set of data, then we define

$$\mathcal{E}_{\text{source}}^i = \{0.9\Sigma_m \leq \hat{\Sigma}_m^i \leq 1.1\Sigma_m, \forall m \in [M]\}$$

Again, as long as $n_m^i \gg d + \log(M\Gamma/\delta), \forall m \in [M], \forall i \in [\Gamma]$, we have

$$\text{Prob}\left(\bigcup_{i \in [\Gamma]} \mathcal{E}_{\text{source}}^i\right) \geq 1 - \frac{\delta}{5}$$

Notice $\mathcal{E}_{\text{target1}}$ will only be used in analyze the main active learning Algorithm 2 while the other two is used in both Algorithm 1 and 2.

C.2. Claims guarantees for unequal sample numbers from source tasks

Here we restate two claims and one result (not written in claim) from (Du et al., 2020), and prove that they still hold when the number of samples drawn from each of the source tasks are not equal, as long as the general low-dimension linear representation is satisfied as defined in Definition 2.1. No benign setting like Definition 2.2 is required.

Algorithm 3 General sample procedure

1: For each task m, draw n_m i.i.d samples from the corresponding offline dataset denoted as $\{X_m, Y_m\}_{m=1}^M$
2: Estimate the models as

$$\hat{\phi}, \hat{W} = \underset{\phi \in \Phi, \hat{W} = [\hat{w}_1, \hat{w}_2, \ldots]}{\arg\min} \sum_{m=1}^M \|\phi(X_m)\hat{w}_m - Y_m\|^2$$

$$\hat{w}_{M+1} = \underset{w}{\arg\min} \|\hat{\phi}(X_{M+1}^i)w - Y_{M+1}\|^2$$

Specidically, consider the above procedure, we show that the following holds for any $\{n_m\}_{m=1}^M$.

Claim C.1 (Modified version of Claim A.3 in (Du et al., 2020)). Given $\mathcal{E}_{\text{source}}$, with probability at least $1 - \delta/10$,

$$\sum_m \|X_m \hat{\Delta}_m\|_2^2 \leq \sigma^2 \left(KM + Kd\log(\kappa(\sum_m n_m)/M) + \log(1/\delta) \right)$$

Proof. We follow nearly the same steps as the proof in (Du et al., 2020), so some details are skipped and we only focus on the main steps that require modification. Also we directly borrow some notations including $\overline{V}, \mathcal{N}, r$ from the original proof and will restate some of them here for clarity.

Notation restatement Since $\text{rank}(\hat{\Delta}) \leq 2K$, we can write $\Delta = VR = [V\boldsymbol{r}_1, \cdots, V\boldsymbol{r}_M]$ where $V \in \mathcal{O}_{d,2K}$ and $R = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_M] \in \mathbb{R}^{2K \times M}$. Here $\mathcal{O}_{d_1,d_2}$ $(d_1 \geq d_2)$ is the set of orthonormal $d_1 \times d_2$ matrices (i.e., the columns are orthonormal). For each $m \in [M]$ we further write $X_m V = U_m Q_m$ where $U_m \in \mathcal{O}_{n_1, 2K}$ and $Q_m \in \mathbb{R}^{2K \times 2K}$. To cover all possible V, we use an ϵ-net argument, that is, there exists any fixed $\overline{V} \in \mathcal{O}_{d,2K}$ and there exists an ϵ-net $\mathcal{N}_\epsilon$ of $\mathcal{O}_{d,2K}$ in Frobenius norm such that $\mathcal{N} \subset \mathcal{O}_{d,2K}$ and $|\mathcal{N}_\epsilon| \leq \left(\frac{6\sqrt{2K}}{\epsilon}\right)^{2Kd}$. (Please refer to original proof for why such ϵ-net exists)

Now we briefly state the proofs.

Step 1:

$$\sum_m \|X_m(\hat{B}\hat{w}_m - B^* w_m^*)\|_2^2 \leq \sum_{m=1}^M \langle Z_m, X_m \overline{V}_m r_m \rangle + \sum_{m=1}^M \langle Z_m, X_m(V - \overline{V}_m)r_m \rangle$$

This comes from the first three lines of step 4 in original proof. For the first term, with probability $1 - \delta/10$, by using standard tail bound for χ^2 random variables and the ϵ-net argument (details in eqn.(28) in original proof), we have it upper bounded by

$$\sigma\sqrt{KM + \log(|\mathcal{N}_\epsilon|/\delta)}\sqrt{\sum_m^M \|X_m V r_m\|_2^2} + \sigma\sqrt{KM + \log(|\mathcal{N}_\epsilon|/\delta)}\sqrt{\sum_m^M \|X_m(\overline{V} - V)r_m\|_2^2}$$

And for the second term, since $\sigma^{-2} \sum_m \|Z_m\|^2 \sim \chi^2\left(\sum_{m=1}^M n_m\right)$, again by using standard tail bound (details in eqn.(29) in original proof), we have that with high probability $1 - \delta/20$, it is upper bounded by

$$\sum_{m=1}^M \langle Z_m, X_m(V - \overline{V}_m)r_m \rangle \lesssim \sigma\sqrt{\sum_{m=1}^M n_m + \log(1/\delta)}\sqrt{\sum_m \|X_m(V - \overline{V}_m)r_m\|_2^2}$$

Step 2: Now we can further bound the term $\sqrt{\sum_m \|X_m(V - \overline{V}_m)r_m\|_2^2}$ by showing

$$\sum_m \|X_m(V - \overline{V}_m)r_m\|_2^2 \le \sum_m \|X_m\|_F^2 \|V - \overline{V}\|_F^2 \|r_m\|_2^2$$

$$\le 1.1\overline{\lambda} \sum_m n_m \|V - \overline{V}\|_F^2 \|r_m\|_2^2$$

$$\le 1.1\overline{\lambda}\epsilon^2 \sum_m n_m \|r_m\|_2^2$$

$$\le 1.1\overline{\lambda}\epsilon^2 \sum_m n_m \|\hat{\Delta}_m\|_2^2$$

$$\le 1.1\kappa\epsilon^2 \sum_m \|X_m(\hat{\Delta}_m)\|_2^2$$

Note this proof is the combination of step 2 and step 3 in the original proof. The only difference here is n_m is different for each m so you need to be more careful on those upper and lower bounds.

Step 3: Finally, we again use the self-bounding techniques. Recall that we have

$$\sqrt{\sum_m \|X_m(\hat{\Delta}_m)\|_2^2} \le \sqrt{\sum_m^M \|Z_m\|_2^2} \sqrt{\sum_m^M \|X_m(\hat{\Delta}_m)\|_2^2}$$

By rearranging the inequality and the distribution of Z_m, we have

$$\sum_m \|X_m(\hat{\Delta}_m)\|_2^2 \le \sigma^2 \left(\sum_{m=1}^M n_m + \log(1/\delta) \right)$$

Step 4: Now replace these into the inequality in step 1, we have

$$\sum_m \|X_m \hat{\Delta}_m\|_2^2 \le \sigma \sqrt{KM + \log(|\mathcal{N}_\epsilon|/\delta)} \sqrt{\sum_m \|X_m \hat{\Delta}_m\|_2^2} + 2\epsilon\sigma^2 \left(\sum_m n_m + \log(1/\delta) \right)$$

Then rearrange the inequality and choose proper ϵ, we get the result.

$\square$

Claim C.2 (Modified version of Claim A.4 in (Du et al., 2020)). Given $\mathcal{E}_{\text{source}}$ and $\mathcal{E}_{\text{target2}}$, with probability at least $1 - \delta/10$,

$$\|P_{X_{M+1}\hat{B}}^\perp X_{M+1} B^* \widetilde{W}^*\|_F^2 \le 1.3 n_{M+1} \sigma^2 \left(KT + Kd \log((\kappa \sum_m n_m)/M) + \log \frac{1}{\delta} \right)$$

where $\widetilde{W}^* = W^* \sqrt{\text{diag}([n_1, n_2, \ldots, n_M])}$

Proof. The proof is almost the same as the first part of the proof except we don't need to extract n_m out.

$$\sum_m \|X_m(\hat{B}\hat{w}_m - B^* w_m^*)\|_2^2 \ge \sum_m \|P_{X_m\hat{B}}^\perp X_m B^* w_m^*\|^2$$

$$\ge 0.9 \sum_m n_m \|P_{\Sigma_m\hat{B}}^\perp \Sigma_m B^* w_m^*\|^2$$

$$= 0.9 \sum_m n_m \|P_{\Sigma_{M+1}\hat{B}}^\perp \Sigma_{M+1} B^* w_m^*\|^2$$

$$= 0.9 \|P_{\Sigma_{M+1}\hat{B}}^\perp \Sigma_{M+1} B^* \tilde{W}^*\|^2$$

$$\ge \frac{0.9}{1.1} \frac{1}{n_{M+1}} \|P_{X_{M+1}\hat{B}}^\perp X_{M+1} B^* \widetilde{W}^*\|_F^2$$

where the first and second inequality is the same as original proof. The third equation comes from our assumption that all Σ_m are the same and the forth equation is just another form of the above term. The last inequality comes from the same reason as second equality, which can again be found in original proof.

Now by using Claim C.1 as an upper bound, we get our desired result.

Basically, this claim is just another way to write Claim A.4 in (Du et al., 2020). Here we combined the n_m with W^* and in original proof they extract n_m can lower bound W^* with its minimum singular value since in their case all n_m are the same. $\qquad\square$

Claim C.3. Given $\mathcal{E}_{\text{source}}$ and $\mathcal{E}_{\text{target2}}$, with probability at least $1 - \delta/5$,

$$\text{ER}\left(\hat{B}, \hat{\boldsymbol{w}}_{M+1}\right) \leq \frac{1}{n_{M+1}} \left\|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^* \boldsymbol{w}^*_{M+1}\right\|_F^2 + \sigma^2 \frac{K + \log(1/\delta)}{n_{M+1}}.$$

Proof. This bound comes exactly from some part of *Proof of Theorem 4.1*. Nothing need to change. $\qquad\square$

D. Analysis for Warm-up

D.1. Proof for Theorem 3.2

Suppose event $\mathcal{E}_{\text{source}}$ and $\mathcal{E}_{\text{target2}}$ holds, then we have with probability at least $1 - \frac{3\delta}{10}$,

$$\begin{aligned}
\text{ER}(\hat{B}, \hat{w}_{M+1}) &\leq \frac{\|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^* w^*_{M+1}\|^2}{n_{M+1}} + \sigma^2 \frac{K + \log(1/\delta)}{n_{M+1}} \\
&= \frac{1}{n_{M+1}} \|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^* \widetilde{W}^* \tilde{\nu}^*\|_2^2 + \sigma^2 \frac{K + \log(1/\delta)}{n_{M+1}} \\
&\leq \frac{1}{n_{M+1}} \|P^{\perp}_{X_{M+1}\hat{B}} X_{M+1} B^* \widetilde{W}^*\|_F^2 \|\tilde{\nu}^*\|_2^2 + \sigma^2 \frac{K + \log(1/\delta)}{n_{M+1}} \\
&= 1.3\sigma^2 \left(KT + Kd\log((\kappa \sum_m n_m)/M) + \log\frac{1}{\delta}\right) \|\tilde{\nu}^*\|_2^2 + \sigma^2 \frac{K + \log(1/\delta)}{n_{M+1}}
\end{aligned}$$

where $\hat{\nu}^*(m) = \frac{\nu^*(m)}{\sqrt{n_m}}$. Here the first inequality comes from Claim C.3 and the last inequality comes from Claim C.2. By use both claim, we have probability at least $1 - \frac{\delta}{10} - \frac{\delta}{5}$. The third inequality comes from holder's inequality.

The key step to our analysis is to decompose and upper bound $\|\tilde{\nu}^*\|_2^2$. Denote $\epsilon^{-2} = \frac{N_{\text{total}}}{\|\nu^*\|_2^2}$, we have, for any $\gamma \in [0, 1]$,

$$\begin{aligned}
\sum_m \frac{\nu^*(m)^2}{n_m} \left(\mathbf{1}\{|\nu^*(m)| > \sqrt{\gamma}\epsilon\} + \mathbf{1}\{|\nu^*(m)| \leq \sqrt{\gamma}\epsilon\}\right) &\lesssim \sum_m \left(\epsilon^2 \mathbf{1}\{|\nu^*(m)| > \sqrt{\gamma}\epsilon\} + \gamma\epsilon^2 \mathbf{1}\{|\nu^*(m)| \leq \sqrt{\gamma}\epsilon\}\right) \\
&\leq \|\nu\|_{0,\gamma}\epsilon^2 + (M - \|\nu\|_{0,\gamma})\gamma\epsilon^2 \\
&= (1 - \gamma)\|\nu\|_{0,\gamma}\epsilon^2 + M\gamma\epsilon^2 \\
&\leq \frac{\|\nu^*\|_2^2}{N_{\text{total}}} \left((1 - \gamma)\|\nu\|_{0,\gamma} + \gamma M\right)
\end{aligned}$$

where the inequality comes from $n_m \geq \frac{1}{2}(\nu^*(m))^2\epsilon^{-2}$.

Finally, combine this with the probability of $\mathcal{E}_{\text{source}}$ and $\mathcal{E}_{\text{target2}}$, we finish the bound.

D.2. Proof for Theorem 3.3

By the same procedure as before, we again get

$$\text{ER}(\hat{B}, \hat{w}_{M+1}) \leq \sigma^2 \left(KT + Kd\log(\kappa(\sum_m n_m)/M) + \log\frac{1}{\delta}\right) \|\tilde{\nu}^*\|_2^2 + \sigma^2 \frac{K + \log(1/\delta)}{n_{M+1}}$$

Now due to uniform sampling, so we have all $n_m = N_{\text{total}}/M$, which means

$$\|\tilde{\nu}^*\|_2^2 = \|\nu^*\|_2^2 \frac{M}{N_{\text{total}}}.$$

Then we get the result by direction calculation.

E. Analysis for Theorem E.4

E.1. Main analysis

Step 1: We first show that the estimated distribution over tasks $\hat{\nu}_i$ is close to the actual distribution ν^* for any fixed i. Notice that Assumption 2.2 is necessary for the proofs in this part.

Lemma E.1 (Closeness between $\hat{\nu}_i$ and ν^*). *Under the Assumption 2.2, given $\mathcal{E}_{source}^i$ and $\mathcal{E}_{target1}$, for any i, m, as long as $n_{M+1} \geq \frac{2000\epsilon_i^{-1}}{\underline{\sigma}^4}$, we have with probability at least $1 - \delta/10M\Gamma$*

$$|\hat{\nu}_{i+1}(m)| \in \begin{cases} [|\nu^*(m)|/16, 4|\nu^*(m)|] & \text{if } \nu^* \geq \sigma\sqrt{\epsilon_i} \\ [0, 4\sqrt{\epsilon_i}] & \text{if } |\nu^*| \leq \sigma\sqrt{\epsilon_i} \end{cases} \tag{8}$$

We define this conditional event as

$$\mathcal{E}_{relevance}^{i,m} = \left\{ Eqn.(8) \text{ holds } \mid \mathcal{E}_{source}^i, \mathcal{E}_{target1} \right\}$$

Proof. By the definition of ν^* and Lemma E.9, we have the following optimization problems,

$$\hat{\nu}_{i+1} = \arg\min_{\nu} \|\nu\|_2^2$$

$$\text{s.t.} \quad \sum_m \alpha_m \left(\hat{\Sigma}_m^i B^* w_m^* + \frac{1}{n_m^i} \left(X_m^i\right)^\top Z_m \right) \nu(m) = \alpha_{M+1} \left(\hat{\Sigma}_M^i B^* w_{M+1}^* + \frac{1}{n_{M+1}^i} \left(X_{M+1}^i\right)^\top Z_{M+1} \right)$$

$$\nu^* = \arg\min_{\nu} \|\nu\|_2^2$$

$$\text{s.t.} \quad \sum_m w_m^* \nu(m) = w_{M+1}^*$$

where $\alpha_m^i = \left(\hat{B}_i^\top \hat{\Sigma}_m^i \hat{B}_i \right)^{-1} \hat{B}_i^\top$.

Now we are ready to show that $\hat{\nu}_{i+1}$ is close to ν^* by comparing their closed form solution.

First by using the lemma E.8 based on the standard KKT condition, it is easy to get a closed form solution of ν^*, that is, for any m.

$$\nu^*(m) = (w_m^*)^\top \left(W^*(W^*)^\top \right)^{-1} w_{M+1}^*$$
$$= (B^* w_m^*)^\top \left(B^* W^* (B^* W^*)^\top \right)^+ (B^* w_{M+1}^*)$$

where the last inequality comes from the fact that B^* is orthonormal. And,

$$
\begin{aligned}
|\hat{\nu}_{i+1}(m)| &= \left| \left((\hat{\Sigma}_m^i B^* w_m^*)^\top + \frac{1}{n_m^i} (Z_m^i)^\top X_m^i \right) \alpha_m^\top (\hat{W}_i \hat{W}_i^\top)^\dagger \alpha_{M+1} \left(\hat{\Sigma}_{M+1}^i B^* w_{M+1}^* + \frac{1}{n_{M+1}^i} (X_{M+1}^i)^\top Z_{M+1}^i \right) \right| \\
&\leq 1.7 \left| (B^* w_m^*)^\top \hat{B}_i (\hat{B}_i^\top \hat{B}_i)^{-1} (\hat{W}_i \hat{W}_i^\top)^\dagger (\hat{B}_i^\top \hat{B}_i)^{-1} \hat{B}_i^\top B^* w_{M+1}^* \right| \\
&\quad + 1.3 \left| \frac{1}{n_m^i} (Z_m^i)^\top X_m^i \hat{B}_i (\hat{B}_i^\top \hat{B}_i)^{-1} (\hat{W}_i \hat{W}_i^\top)^\dagger (\hat{B}_i^\top \hat{B}_i)^{-1} \hat{B}_i^\top B^* w_{M+1}^* \right| \\
&\quad + 1.3 \left| (B^* w_m^*)^\top \hat{B}_i (\hat{B}_i^\top \hat{B}_i)^{-1} (\hat{W}_i \hat{W}_i^\top)^\dagger (\hat{B}_i^\top \hat{B}_i)^{-1} \hat{B}_i^\top \frac{1}{n_{M+1}^i} (X_{M+1}^i)^\top Z_{M+1}^i \right| \\
&\quad + \left| \frac{1}{n_m^i} (Z_m^i)^\top X_m^i \hat{B}_i (\hat{B}_i^\top \hat{B}_i)^{-1} (\hat{W}_i \hat{W}_i^\top)^\dagger (\hat{B}_i^\top \hat{B}_i)^{-1} \hat{B}_i^\top \frac{1}{n_{M+1}^i} (X_{M+1}^i)^\top Z_{M+1}^i \right| \\
&\leq 1.7 \left| (B^* w_m^*)^\top \hat{B}_i (\hat{W}_i \hat{W}_i^\top)^\dagger \hat{B}_i^\top B^* w_{M+1}^* \right| \\
&\quad + 1.3 \left| \frac{1}{n_m^i} (Z_m^i)^\top X_m^i \hat{B}_i (\hat{W}_i \hat{W}_i^\top)^\dagger \hat{B}_i^\top B^* w_{M+1}^* \right| \\
&\quad + 1.3 \left| (B^* w_m^*)^\top \hat{B}_i (\hat{W}_i \hat{W}_i^\top)^\dagger \hat{B}_i^\top \frac{1}{n_{M+1}^i} (X_{M+1}^i)^\top Z_{M+1}^i \right| \\
&\quad + \left| \frac{1}{n_m^i} (Z_m^i)^\top X_m^i \hat{B}_i (\hat{W}_i \hat{W}_i^\top)^\dagger \hat{B}_i^\top \frac{1}{n_{M+1}^i} (X_{M+1}^i)^\top Z_{M+1}^i \right| \\
&\leq 2 \left| (B^* w_m^*)^\top (\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^\top)^\dagger B^* w_{M+1}^* \right| + \textit{noise term}(m)
\end{aligned}
$$

The first inequality comes from the definition of α_m and the event $\mathcal{E}_{\text{source}}, \mathcal{E}_{\text{target1}}$. The second equality comes from that $\hat{B}_i$ are always orthonormal matrix. Notice that in practice this is not required. Finally we set the last three terms in the second inequality as *noise term*(m), which is a low order term that we will shown later.

—-Sub-step 1 (Analyze the non noise-term): We have the difference between $|\hat{\nu}_{i+1}(m)|$ and $2|\nu^*(m)|$ is

$$
\begin{aligned}
&|\hat{\nu}_{i+1}(m)| - 2|\nu^*(m)| - \textit{noise term}(m) \\
&\leq 2 \left| (B^* w_m^*)^\top (\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^\top)^\dagger B^* w_{M+1}^* \right| - 2 \left| (B^* w_m^*)^\top \left(B^* W^* (B^* W^*)^\top \right)^+ (B^* w_{M+1}^*) \right| \\
&\leq 2 \left| (B^* w_m^*)^\top \left((\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^\top)^\dagger - \left(B^* W^* (B^* W^*)^\top \right)^+ \right) (B^* w_{M+1}^*) \right| \\
&\leq 2 \left| (B^* w_m^*)^\top \left(B^* W^* (B^* W^*)^\top \right)^\dagger \left(\hat{\Delta}^i \hat{\Delta}_i^\top + \hat{\Delta}^i (B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) \left(B^* W^* (B^* W^*)^\top \right)^\dagger (B^* w_{M+1}^*) \right| \\
&\leq 2 \| B^* w_m^* \|_2 \| B^* w_{M+1}^* \|_2 \| \left(B^* W^* (B^* W^*)^\top \right)^\dagger \left(\hat{\Delta}^i (\hat{\Delta}^i)^\top + \hat{\Delta}^i (B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) \left(B^* W^* (B^* W^*)^\top \right)^\dagger \|_F \\
&\leq 2R \| \left(B^* W^* (B^* W^*)^\top \right)^\dagger \left(\hat{\Delta}^i (\hat{\Delta}^i)^\top + \hat{\Delta}^i (B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) \left(B^* W^* (B^* W^*)^\top \right)^\dagger \|_F \\
&\leq \sigma \sqrt{\epsilon_i} + \textit{noise term}(m)
\end{aligned}
$$

where the second inequality comes from the triangle inequality, the third inequality holds with probability at least $1 - \delta$ by using the application of generalized inverse of matrices theorem (see Lemma E.10 for details) and the fifth inequality comes form the fact that $\| B^* w_m^* \|_2 \leq \| w_m^* \|_2 \leq R$. Finally, by using the assumptions of B^*, W^* as well as the multi-task model estimation error $\hat{\Delta}^i$, we can apply Lemma E.12 to get the last inequality.

With the same reason, we get that, for the other direction,

$$
0.5 |\nu^*(m)| - |\hat{\nu}_{i+1}(m)| - \textit{noise term}(m) \leq \sigma \sqrt{\epsilon_i}/4
$$

Combine these two, we have

$$
|\hat{\nu}_{i+1}(m)| \in [0.5|\nu^*(m)| - \sigma\sqrt{\epsilon_i}/4 - 1.5 \textit{noise term}(m) \quad , \quad 2|\nu^*(m)| + \sigma\sqrt{\epsilon_i} + 1.5 \textit{noise term}(m)]
$$

—-**Sub-step 2 (Analyze the noise-term):** Now let's deal with the *noise term*(m), we restate it below for convenience,

$$1.3\left|\frac{1}{n_m^i}(Z_m^i)^\top X_m^i \hat{B}_i(\hat{W}_i\hat{W}_i^\top)^\dagger \hat{B}_i^\top B^* w_{M+1}^*\right| + 1.3\left|(B^* w_m^*)^\top \hat{B}_i(\hat{W}_i\hat{W}_i^\top)^\dagger \hat{B}_i^\top \frac{1}{n_{M+1}^i}(X_{M+1}^i)^\top Z_{M+1}^i\right|$$

$$+ \left|\frac{1}{n_m^i}(Z_m^i)^\top X_m^i \hat{B}_i(\hat{W}_i\hat{W}_i^\top)^\dagger \hat{B}_i^\top \frac{1}{n_{M+1}^i}(X_{M+1}^i)^\top Z_{M+1}^i\right|.$$

By the assumption on B^*, W^*, it is easy to see that with high probability at least $1 - \delta'$, where $\delta' = \delta/10\Gamma M$

$$\left|\frac{1}{n_m^i}(Z_m^i)^\top X_m^i (\hat{B}_i\hat{W}_i(\hat{B}_i\hat{W}_i)^\top)^\dagger B^* w_{M+1}^*\right|$$

$$\leq \left|\frac{1}{\lambda_{\min}(B^*W^*(B^*W^*)^\top)}\frac{1}{n_m^i}(Z_m^i)^\top X_m^i B^* w_{M+1}^*\right|$$

$$\leq \frac{\sigma}{n_m^i \lambda_{\min}(B^*W^*(B^*W^*)^\top)}\sqrt{(w_{M+1}^*)^\top (B^*)^\top (X_m^i)^\top X_m^i B^* w_{M+1}^* \log(1/\delta')}$$

$$\leq \frac{2.2\sigma\|w_{M+1}^*\|_2\sqrt{\log(1/\delta')}}{\sqrt{n_m^i}\lambda_{\min}(B^*W^*(B^*W^*)^\top)}$$

$$\leq \sqrt{\epsilon_i/\beta}\frac{2.2\sigma\sqrt{R\log(1/\delta')}}{\lambda_{\min}(B^*W^*(B^*W^*)^\top)}$$

where the first inequality comes from Lemma E.6, the second the inequality comes from Chernoff inequality and the last inequality comes from the definition $n_m^i = \max\{\beta\hat{\nu}_i^2\epsilon_i^{-2}, \beta\epsilon_i^{-1}, \underline{N}\}$. Note that we choose $\beta = 3000K^2R^2(KM + Kd\log(1/\varepsilon M) + \log(M\Gamma/\delta)/\underline{\sigma}^6$. Therefore, above can be upper bounded by $\sqrt{\epsilon_i}/24$.

By the similar argument and the assumption that $n_{M+1} \geq \frac{3000R\epsilon_i^{-1}}{\underline{\sigma}^4}$, we can also show that with high probability at least $1 - \delta'$

$$\left|(B^* w_m^*)^\top \hat{B}_i(\hat{W}_i\hat{W}_i^\top)^\dagger \hat{B}_i^\top \frac{1}{n_{M+1}^i}(X_{M+1}^i)^\top Z_{M+1}^i\right| \leq \frac{2.2\sigma\|w_m^*\|_2\sqrt{\log(1/\delta')}}{\sqrt{n_{M+1}}\lambda_{\min}(B^*W^*(B^*W^*)^\top)}$$

$$\leq \sigma\sqrt{\epsilon_i}/24$$

Finally, we have that

$$\left|\frac{1}{n_m^i}(Z_m^i)^\top X_m^i \hat{B}_i(\hat{W}_i\hat{W}_i^\top)^\dagger \hat{B}_i^\top \frac{1}{n_{M+1}^i}(X_{M+1}^i)^\top Z_{M+1}^i\right| \leq \frac{2.2\sigma^2\sqrt{\epsilon_i/\beta}\|w_m^*\|_2\|w_{M+1}^*\|_2\log(1/\delta')}{\sqrt{n_{M+1}}\lambda_{\min}(B^*W^*(B^*W^*)^\top)}$$

$$\leq \sigma\sqrt{\epsilon_i}/24$$

So overall we have we have *noise term*(m) $\leq \sigma\sqrt{\epsilon_i}/8$.

—-**Sub-step 3 (Combine the non noise-term and noise-term):**

Now when $|\nu^*(m)| \geq \sigma\sqrt{\epsilon_i}$, combine the above results show that $|\hat{\nu}_{i+1}(m)| \in [\nu^*(m)/16, 4|\nu^*(m)|]$.

On the other hand, if $|\nu^*| \leq \sigma\sqrt{\epsilon_i}$, then we directly have $\hat{\nu}_{i+1} \in [0, 4\sigma\sqrt{\epsilon_i}]$.

$\square$

Step 2: Now we are ready to prove the following two main lemmas on the final accuracy and the total sample complexity.

Lemma E.2 (Accuracy on each epoch). *Given $\mathcal{E}_{source}, \mathcal{E}_{target\,1}, \mathcal{E}_{target\,2}$ and $\mathcal{E}_{relevance}^{i,m}$ for all m, after the epoch i, we have* $\mathrm{ER}(\hat{B}, \hat{w}_{M+1})$ *upper bounded by with probability at least $1 - \delta/10M\Gamma$.*

$$\frac{\sigma^2}{\beta}\left(KM + Kd + \log\frac{1}{\delta}\right)s_i^*\epsilon_i^2 + \sigma^2\frac{(K + \log(1/\delta))}{n_{M+1}}$$

where $s_i^* = \min_{\gamma\in[0,1]}(1-\gamma)\|\nu^*\|_{0,\gamma}^i + \gamma M$ and $\|\nu\|_{0,\gamma}^i := |\{m : \nu_m > \sqrt{\gamma}\epsilon_i\}|$.

Proof. The first step is the same as proof of Theorem 3.2 in Appendix 3. Suppose event $\mathcal{E}_{\text{source}}^i$ and $\mathcal{E}_{\text{target2}}$ holds, then we have with probability at least $1 - \frac{3\delta}{10\Gamma M}$,

$$\text{ER}(\hat{B}, \hat{w}_{M+1}) \leq 1.3\sigma^2 \left(KT + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta} \right) \|\tilde{\nu}^*\|_2^2 + \sigma^2 \frac{K + \log(1/\delta)}{n_{M+1}}$$

where $\tilde{\nu}^*(m) = \nu^*(m)/\sqrt{n_m}$.

Now for any $\gamma \in [0, 1]$, given $\mathcal{E}_{\text{relevance}}^{i,m}$, we are going to bound $\sum_m \frac{\nu^*(m)^2}{n_m^i}$ as

$$\sum_m \frac{\nu^*(m)^2}{n_m^i} \leq \sum_m \frac{\nu^*(m)^2}{n_m^i} \mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\} + \sum_m \frac{\nu^*(m)^2}{n_m^i} \mathbf{1}\{\sqrt{\gamma}\epsilon_{i-1} \leq |\nu^*(m)| \leq \sigma\sqrt{\epsilon_{i-1}}\}$$

$$+ \sum_m \frac{\nu^*(m)^2}{n_m^i} \mathbf{1}\{|\nu^*(m)| \leq \sqrt{\gamma}\epsilon_i\}$$

$$\leq \sum_m \frac{256\hat{\nu}_i^2}{n_m^i} \mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\} + \sum_m \frac{\sigma^2\epsilon_{i-1}}{n_m^i} \mathbf{1}\{\sqrt{\gamma}\epsilon_{i-1} \leq |\nu^*(m)| \leq \sqrt{\epsilon_{i-1}}\}$$

$$+ \gamma\epsilon_i^2/\beta \sum_m \mathbf{1}\{|\nu^*(m)| \leq \sqrt{\gamma}\epsilon_i\}$$

$$\leq \mathcal{O}\left(\sum_m \epsilon_i^2/\beta\mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\} \right) + \mathcal{O}\left(\sum_m \sigma^2\epsilon_i^2/\beta\mathbf{1}\{\sqrt{\gamma}\epsilon_{i-1} \leq |\nu^*(m)| \leq \sigma\sqrt{\epsilon_{i-1}}\} \right)$$

$$+ (M - \|\nu\|_{0,\gamma}^i)\gamma\epsilon_i^2/\beta$$

$$\leq \mathcal{O}\left(\sum_m \epsilon_i^2/\beta\mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\gamma}\epsilon_{i-1}\} \right) + (M - \|\nu\|_{0,\gamma}^i)\gamma\epsilon_i^2/\beta$$

$$\leq ((1-\gamma)\|\nu\|_{0,\gamma}^i + \gamma M)\epsilon_i^2/\beta$$

$\square$

Lemma E.3 (Sample complexity on each epoch). *Given $\mathcal{E}_{\text{source}}, \mathcal{E}_{\text{target 1}}$ and $\mathcal{E}_{\text{relevance}}^{i,m}$ for all m, we have the total number of source samples used in epoch i as as*

$$\mathcal{O}\left(\beta(M\varepsilon^{-1} + \|\nu^*\|_2^2\varepsilon^{-2}) \right)$$

Proof. Given $\mathcal{E}_{\text{relevance}}^{i,m}$, we can get the sample complexity for block i as the follows.

$$\sum_m^M n_m^i = \sum_m^M \max\{\hat{\nu}_i(m)^2\epsilon_i^{-2}, \epsilon_i^{-1}, \underline{N}\}$$

$$\leq \sum_m^M \beta\hat{\nu}_i^2\epsilon_i^{-2} + \sum_m^M \beta\epsilon_i^{-1}$$

$$\leq \sum_m^M \beta\hat{\nu}_i^2(m)\epsilon_i^{-2}\mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\} + \sum_m^M \beta\hat{\nu}_i^2(m)\epsilon_i^{-2}\mathbf{1}\{|\nu^*(m)| \leq \sigma\sqrt{\epsilon_{i-1}}\} + \sum_m^M \beta\epsilon_i^{-1}$$

$$\leq \sum_m^M \beta(4\nu^*(m))^2\epsilon_i^{-2}\mathbf{1}\{|\nu^*(m)| > \sigma\sqrt{\epsilon_{i-1}}\} + \sum_m^M \beta(4\sigma\sqrt{\epsilon_{i-1}})^2/\epsilon_i^{-2}\mathbf{1}\{|\nu^*(m)| \leq \sigma\sqrt{\epsilon_{i-1}}\} + \sum_m^M \beta\epsilon_i^{-1}$$

$$= \mathcal{O}\left(\beta(M\epsilon_i^{-1} + \|\nu^*\|_2^2\epsilon_i^{-2}) \right)$$

$\square$

Theorem E.4. *Suppose we know in advance a lower bound of $\sigma_{\min}(W^*)$ denoted as $\underline{\sigma}$. Under the benign low-dimension linear representation setting as defined in Assumption 2.2, we have $\mathrm{ER}(\hat{B}, \hat{w}_{M+1}) \le \varepsilon^2$ with probability at least $1 - \delta$ whenever the number of source samples N_{total} is at least*

$$\widetilde{\mathcal{O}}\left(\left(K(M+d) + \log\frac{1}{\delta} \right) \sigma^2 s^* \|\nu^*\|_2^2 \varepsilon^{-2} + \Box\sigma\varepsilon^{-1} \right)$$

where $\Box = \left(MK^2 dR/\underline{\sigma}^3 \right)\sqrt{s^*}$ and the target task sample complexity n_{M+1} is at least

$$\widetilde{\mathcal{O}}\left(\sigma^2 K\varepsilon^{-2} + \Diamond\sqrt{s^*}\sigma\varepsilon^{-1} \right)$$

where $\Diamond = \min\left\{ \frac{\sqrt{R}}{\underline{\sigma}^2 K}, \sqrt{K(M+d) + \log\frac{1}{\delta}} \right\}$ and s^* has been defined in Theorem 3.2.

Proof. Given $\mathcal{E}_{\text{source}}, \mathcal{E}_{\text{target 1}}, \mathcal{E}_{\text{target 2}}$ and $\mathcal{E}_{\text{relevance}}^{i,m}$ then by Lemma E.2, we the final accuracy from last epoch as

$$\mathrm{ER}_{M+1}(\hat{B}, \hat{w}_{M+1}) \le \sigma^2 \left(KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta} \right) s_\Gamma^* \epsilon_\Gamma^2/\beta + \sigma^2\frac{(K + \log(1/\delta))}{n_{M+1}}$$

Denote the final accuracy of the first term as ε^2. So we can write ϵ_Γ as

$$\varepsilon/\left(\sigma\sqrt{KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta}}\sqrt{s_\Gamma^*/\beta} \right)$$

By applying lemma 4.4, we requires the total source sample complexity

$$\sum_{i=1}^{\Gamma} \beta(M\epsilon_i^{-1} + \|\nu^*\|_2^2\epsilon_i^{-2}) \le 2\beta(M\epsilon_\Gamma^{-1} + 2\|\nu^*\|_2^2\epsilon_\Gamma^{-2})$$

$$= \beta M\sigma\sqrt{KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta}}\sqrt{s_\Gamma^*/\beta}\varepsilon^{-1}$$

$$+ \beta\|\nu^*\|_2^2\sigma^2\left(KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta} \right) s_\Gamma^*\varepsilon^{-2}/\beta$$

$$= \sqrt{\beta}M\sqrt{s_\Gamma^*}\sigma\sqrt{KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta}}\varepsilon^{-1}$$

$$+ s_\Gamma^*\sigma^2\left(KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta} \right)\|\nu^*\|_2^2\varepsilon^{-2}$$

$$= \widetilde{\mathcal{O}}\left(\left(MK^2 d + M\sqrt{Kd}/\underline{\sigma}^2 \right)\sqrt{s_\Gamma^*}\sigma\varepsilon^{-1} + Kds_\Gamma^*\sigma^2\|\nu^*\|_2^2\varepsilon^{-2} \right)$$

Also in order to satisfy the assumption in Lemma E.1, we required n_{M+1} to be at least

$$\frac{\varepsilon^{-1}}{\underline{\sigma}^2} = \frac{1}{\underline{\sigma}^2}\sqrt{s_\Gamma^*/\beta}\sigma\sqrt{KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta}}\varepsilon^{-1}$$

$$\le \min\left\{ \frac{1}{\underline{\sigma}^2 K}, \sqrt{KM + Kd\log((\sum_m n_m)/M) + \log\frac{1}{\delta}} \right\}\sqrt{s_\Gamma^*}\sigma\varepsilon^{-1}$$

Notice that Γ is a algorithm-dependent parameter, therefore, the final step is to bound s_Γ^* by an algorithm independent term by write Γ as

$$\Gamma = -\log\epsilon_\Gamma \le \min\left\{ \log\sqrt{\frac{N_{total}}{\beta\|\nu^*\|_2^2}}, \log\frac{N_{total}}{\beta M} \right\}$$

So we have .

$$\|\nu\|_{0,\gamma}^{\Gamma} = |\{m : \nu_m > \sqrt{\gamma} \max\{\frac{\beta\|\nu^*\|_2^2}{N_{total}}, \frac{\beta M}{N_{total}}\}|$$

To further simply this, notice that, for any $\epsilon' < \epsilon_i$.

$$\|\nu\|_{0,\gamma}^{i} = |\{m : \nu_m > \sqrt{\gamma}\epsilon_i\}| \leq |\{m : \nu_m > \sqrt{\gamma}\epsilon'\}|$$

So we further have

$$\|\nu\|_{0,\gamma}^{\Gamma} = |\{m : \nu_m > \sqrt{\gamma}\frac{\|\nu^*\|_2^2}{N_{total}}\}| := \|\nu\|_{0,\gamma}$$

Finally, by union bound $\mathcal{E}_{source}$, $\mathcal{E}_{target\ 1}$, $\mathcal{E}_{target\ 2}$ and $\mathcal{E}_{relevance}^{i,m}$ on all epochs, we show that all the lemmas holds with probability at least $1 - \delta$.

$\square$

E.2. Auxiliary Lemmas

Lemma E.5 (Convergence on estimated model $\hat{B}_i\hat{W}_i$). *For any fixed i, given $\mathcal{E}_{source}^i$, we have*

$$\|\hat{\Delta}^i\|_F^2 \leq 1.3\sigma^2 \left(KM + Kd\log((\sum_m n_m^i)/M) + \log\frac{10\Gamma}{\delta}\right)\epsilon_i/\beta$$

And therefore, when $\beta = 3000K^2R^2(KM + Kd\log(N_{total}/M) + \log(M\Gamma/\delta)/\underline{\sigma}^6$. Therefore, above can be upper bounded by $\sqrt{\epsilon_i}/24$,

we have $\|\Delta\|_F^2 \leq \frac{\sigma^2\epsilon_i}{4K^2R^2}$

Proof. Denote Δ_m as the m-th column of $\hat{\Delta}^i$.

$$\sum_{m=1}^{M} \|X_m\Delta_m\|_2^2 = \sum_{m=1}^{M} \Delta_m^\top X_m^\top X_m \Delta_m$$

$$\geq 0.9 \sum_{m=1}^{M} n_m \Delta_m^\top \Delta_m$$

$$\geq 0.9 \min_m n_m^i \sum_{m=1}^{M} \|\Delta_m\|_2^2 = 0.9 \min_m n_m^i \|\hat{\Delta}^i\|_F^2$$

Recall that our definition on $n_m^i = \max\{\beta\hat{\nu}_i^2(m)\epsilon_i^{-2}, \beta\epsilon_i^{-1}\}$ and also use the upper bound derived in Claim C.1, we finish the proof. $\square$

Lemma E.6 (minimum singular value guarantee for $\hat{B}_i\hat{W}_i$). *For all i, we can guarantee that*

$$\sigma_{min}(\hat{B}_i\hat{W}_i) \geq \sigma_{min}(W^*)/2$$

Also because $M \geq K$, so there is always a feasible solution for $\hat{\nu}_i$.

Proof. Because B^* is a orthonormal, so $\sigma_{min}(B^*W^*) = \sigma_{min}(W^*)$. Also from Lemma E.5 and Weyl's theorem stated below, we have $|\sigma_{min}(\hat{B}_i\hat{W}_i)) - \sigma_{min}(B^*W^*)| \leq \|\hat{B}_i\hat{W}_i - B^*W^*\|_F \leq \frac{\sigma}{2} \leq \frac{\sigma_{min}(W^*)}{2}$. Combine these two inequality we can easily get the result.

Theorem E.7 (Weyl's inequality for singular values). *Let M be a $p \times n$ matrix with $1 \le p \le n$. Its singular values $\sigma_k(M)$ are the p positive eigenvalues of the $(p+n) \times (p+n)$ Hermitian augmented matrix*

$$\begin{bmatrix} 0 & M \\ M^* & 0 \end{bmatrix}$$

Therefore, Weyl's eigenvalue perturbation inequality for Hermitian matrices extends naturally to perturbation of singular values. This result gives the bound for the perturbation in the singular values of a matrix M due to an additive perturbation Δ :

$$|\sigma_k(M + \Delta) - \sigma_k(M)| \le \sigma_1(\Delta) \le \|\Delta\|_F$$

$\square$

Lemma E.8. *For any two matrix $M_1 \in \mathbb{R}^{K \times M}, M_2 \in \mathbb{R}^K$ where $K \le M$. Suppose $\text{rank}(M_1) = K$ and define $\tilde{\nu}$ as*

$$\underset{\nu \in \mathbb{R}^M}{\arg\min} \|\nu\|_2^2 \quad s.t. \ M_1 \nu = M_2, \nu \in \mathbb{R}^m,$$

then we have

$$\tilde{\nu} = M_1^\top (M_1 M_1^\top)^{-1} M_2$$

Proof. We prove this by using KKT conditions,

$$L(\nu, \lambda) = \|\nu\|_2^2 + \lambda^\top (M_1 \nu_1 - M_2)$$

Given $0 \in \partial L$, we have $\tilde{\nu} = -(M_1)^\top \lambda / 2$. Then by replace this into the constrains, we have

$$M_1 M_1^\top \lambda = -2 M_2 \to \lambda = -2 \left(M_1 M_1^\top \right)^{-1} M_2$$

and therefore $\tilde{\nu} = (M_1)^\top \left(M_1 M_1^\top \right)^{-1} M_2$. $\square$

Lemma E.9 (A closed form expression for $\hat{\nu}_i$). *For any epoch i, given the estimated representation $\hat{B}_i$, we have*

$$\hat{\nu}_{i+1} = \underset{\nu}{\arg\min} \|\nu\|_2^2$$

$$s.t. \quad \sum_m \alpha_m^i \left(\hat{\Sigma}_m^i B^* w_m^* + \frac{1}{n_m^i} \left(X_m^i \right)^\top Z_m \right) \nu(m) = \alpha_{M+1}^i \left(\hat{\Sigma}_M^i B^* w_{M+1}^* + \frac{1}{n_{M+1}^i} \left(X_{M+1}^i \right)^\top Z_{M+1} \right)$$

where $\alpha_m^i = \left(\hat{B}_i^\top \hat{\Sigma}_m^i \hat{B}_i \right)^{-1} \hat{B}_i^\top$

Proof. For any epoch i and it's estimated representation $\hat{B}_i$, by least square argument, we have

$$\hat{w}_m^i = \underset{w}{\arg\min} \|X_m^i \hat{B}_i w - Y_m\|_2$$

$$= \left(\left(X_m^i \hat{B}_i \right)^\top X_m^i \hat{B}_i \right)^{-1} \left(X_m^i \hat{B}_i \right)^\top Y_m$$

$$= \left(\left(X_m^i \hat{B}_i \right)^\top X_m^i \hat{B}_i \right)^{-1} \left(X_m^i \hat{B}_i \right)^\top X_m^i B^* w_m^* + \left(\left(X_m^i \hat{B}_i \right)^\top X_m^i \hat{B}_i \right)^{-1} \left(X_m^i \hat{B}_i \right)^\top Z_m$$

$$= \underbrace{\left(\hat{B}_i^\top \hat{\Sigma}_m^i \hat{B}_i \right)^{-1} \hat{B}_i^\top}_{\alpha_m^i \in \mathbb{R}^{k \times d}} \hat{\Sigma}_m^i B^* w_m^* + \underbrace{\left(\hat{B}_i^\top \hat{\Sigma}_m^i \hat{B}_i \right)^{-1} \hat{B}_i^\top}_{\alpha_m^i} \left(X_m^i \right)^\top Z_m$$

Therefore, combine this with the previous optimization problem, we have which implies that

$$\hat{W}_i \nu = \sum_m \alpha_m^i \left(\hat{\Sigma}_m^i B^* w_m^* + \frac{1}{n_m^i} \left(X_m^i \right)^\top Z_m \right) \nu(m)$$

$$\hat{w}_{M+1}^i = \alpha_{M+1}^i \left(\hat{\Sigma}_M^i B^* w_{M+1}^* + \frac{1}{n_{M+1}^i} \left(X_{M+1}^i \right)^\top Z_{M+1} \right)$$

Recall the definition of $\hat{\nu}_{i+1}$ as

$$\min \|\nu\|_2^2 \quad \text{s.t.} \quad \hat{W}_i \nu = \hat{w}_{M+1}^i$$

Therefore we get the closed-form by replace $\hat{W}_i \nu, \hat{w}_{M+1}^i$ with the value calculated above. □

Lemma E.10 (Difference of the inverse covariance matrix). *For any fixed i, m and any proper matrices M_1, M_2, M_3, M_4,*

$$\left| M_1 \left((\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^\top)^\dagger - (B^* W^* (B^* W^*)^\top)^+ \right) B^* M_2 \right|$$

$$= \|M_1\| \| \left(B^* W^* (B^* W^*)^\top \right)^\dagger \left(\hat{\Delta}_i (\hat{\Delta}^i)^\top + \hat{\Delta}^i (B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) (B^* W^* (B^* W^*)^\top)^\dagger \| \|B^* M_2\|$$

$$\left| M_3 (B^*)^\top \left((\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^\top)^\dagger - (B^* W^* (B^* W^*)^\top)^+ \right) M_4 \right|$$

$$= \|M_3 (B^*)^\top\| \| \left(B^* W^* (B^* W^*)^\top \right)^\dagger \left(\hat{\Delta}^i (\hat{\Delta}^i)^\top + \hat{\Delta}(B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) (B^* W^* (B^* W^*)^\top)^\dagger \| \|M_4\|$$

Proof. First we want to related these two inverse term,

$$(\hat{B}_i \hat{W}_i \hat{W}_i^\top \hat{B}_i^\top)^\dagger - \left(B^* W^* (B^* W^*)^\top \right)^\dagger$$

$$\leq \left((B^* W^* + \hat{\Delta}^i)(B^* W^* + \hat{\Delta}_i)^\top \right)^\dagger - \left(B^* W^* (B^* W^*)^\top \right)^\dagger$$

$$\leq \left(B^* W^* (B^* W^*)^\top + \left(\hat{\Delta}^i (\hat{\Delta}^i)^\top + \hat{\Delta}^i (B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) \right)^\dagger - \left(B^* W^* (B^* W^*)^\top \right)^\dagger$$

In order to connect the first pseudo inverse with the second, we want to use the generalized inverse of matrix theorem as stated below.

Theorem E.11 (Theorem from (Kovanic, 1979)). *If V is an $n \times n$ symmetrical matrix and if X is an $n \times q$ arbitrary real matrix, then*

$$(V + XX^\top)^\dagger = V^\dagger - V^\dagger X(I + X^\top V^\dagger X)^{-1} X^\top V^\dagger + ((X_\perp)^\dagger)^\top X_\perp^\dagger$$

where $X_\perp = (I - VV^\dagger)X$

It is easy to see that $V := B^* W^* (B^* W^*)^\top$ and we can also decompose $\left(\hat{\Delta} \hat{\Delta}^\top + \hat{\Delta}(B^* W^*)^\top + (B^* W^*)\hat{\Delta}^\top \right)$ into some $XX^\top$,

Therefore, we can write the above inequality as

$$-V^\dagger X(I + X^\top V^\dagger X)^{-1} X^\top V^\dagger + ((X_\perp)^\dagger)^\top X_\perp^\dagger$$

Next we show that $((X_\perp)^\dagger)^\top X_\perp^\dagger B^* = 0$ and $(B^*)^\top ((X_\perp)^\dagger)^\top X_\perp^\dagger = 0$. Let $UDV^\top$ be the singular value decomposition of $B^* W^*$. So we have

$$VV^\dagger = UD^2 U^\top (UD^2 U^\top)^\dagger = UU^\top,$$

and therefore, because B^* are contained in the column spaces as $B^* W^*$,

$$X_\perp^\dagger B^* = (U_\perp U_\perp^\top X)^\dagger B^*$$
$$= X^\dagger U_\perp U_\perp^\top B^* = 0.$$

Therefore, we conclude that

$$\left| M_1 \left((\hat{B}_i \hat{W}_i (\hat{B}_i \hat{W}_i)^\top)^\dagger - (B^* W^* (B^* W^*)^\top)^+ \right) B^* M_2 \right|$$

$$\leq \|M_1\| \| \underbrace{\left(B^* W^* (B^* W^*)^\top \right)^\dagger \left(\hat{\Delta} (\hat{\Delta}^i)^\top + \hat{\Delta} (B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) (B^* W^* (B^* W^*)^\top)^\dagger}_{V^\dagger X X^\top V^\dagger} \| \| B^* M_2 \|$$

and so does the other equation. $\qquad\square$

Lemma E.12. *Given $\mathcal{E}_{source}^i$, for any fixed i, we have*

$$\| \left(B^* W^* (B^* W^*)^\top \right)^\dagger \left(\hat{\Delta}^i (\hat{\Delta}^i)^\top + \hat{\Delta} (B^* W^*)^\top + (B^* W^*)(\hat{\Delta}^i)^\top \right) (B^* W^* (B^* W^*)^\top)^\dagger \|_F$$

$$\leq 3\sigma \|(W^*)^\dagger\|_F \|(W^*(W^*)^\top)^\dagger\|_F \underline{\sigma}^3 \sqrt{\epsilon_i} / 6KR$$

$$\leq \sigma \sqrt{\epsilon_i} / 2R$$

Proof. The target term can be upper bounded by

$$\| \left(B^* W^* (B^* W^*)^\top \right)^\dagger \hat{\Delta}^i (\hat{\Delta}^i)^\top (B^* W^* (B^* W^*)^\top)^\dagger \|_F$$

$$+ \| \left(B^* W^* (B^* W^*)^\top \right)^\dagger B^* W^* (\hat{\Delta}^i)^\top (B^* W^* (B^* W^*)^\top)^\dagger \|_F$$

$$+ \| \left(B^* W^* (B^* W^*)^\top \right)^\dagger \hat{\Delta}^i (B^* W^*)^\top (B^* W^* (B^* W^*)^\top)^\dagger \|_F$$

Before we do the final bounding, we first show the upper bound of the following term, which will be used a lot,

$$\begin{aligned}
\|(B^* W^*)^\top \left(B^* W^* (B^* W^*)^\top \right)^\dagger \|_F &= \| \left(B^* W^* (B^* W^*)^\top \right)^\dagger B^* W^* \|_F \\
&= \|(B^* W^* (W^*)^\top (B^*)^\top)^\dagger B^* W^* \|_F \\
&= \|((B^*)^\top)^\dagger (W^*(W^*)^\top)^\dagger (B^*)^\dagger B^* W^* \|_F \\
&= \|B^* (W^*(W^*)^\top)^\dagger W^* \|_F \\
&= \|(W^*(W^*)^\top)^\dagger W^* \|_F \\
&= \|(W^*)^\top)^\dagger (W^*)^\dagger W^* \|_F \\
&= \|((W^*)^\dagger W^*)^\top (W^*)^\dagger \|_F \\
&= \|(W^*)^\dagger W^* (W^*)^\dagger \|_F \\
&= \|((W^*)^\dagger \|_F
\end{aligned}$$

Therefore, we can bound the whole term by

$$\|(B^* W^* (B^* W^*)^\top)^\dagger\|_F^2 \|\hat{\Delta}^i\|_F^2 + 2\|(W^*)^\dagger\|_F \|\hat{\Delta}^i\|_F \|(B^* W^* (B^* W^*)^\top)^\dagger\|_F \leq 3\|(W^*)^\dagger\| \|(W^*(W^*)^\top)^\dagger\|_F \|\hat{\Delta}^i\|_F$$

Recall that we have $\|\hat{\Delta}^i\|_F^2 \leq \frac{\sigma^2 \underline{\sigma}^6 \epsilon_i}{36K^2R^2}$ given $\mathcal{E}_{source}^i$ therefore, we get the final result. $\qquad\square$

F. Experiment details

F.1. Other implementation details

We choose $\beta_i = 1/\|\nu\|_2^2$, which is usually $\Theta(1)$ in practice. Instead of choosing $\epsilon_i = 2^i$, we set that as 1.5^{-i} and directly start from $i = 22$. It is easy to see that it turns out the actual sample number used in the experiment is similar as choosing $\beta = \text{poly}(d, K, M)$ and start from epoch 1 as proposed in theorem. But our choice is more easy for us to adjust parameter and do comparison.

We run each experiment on each task only once due to the limited computational resources and admitted that it is better to repeat the experiment for more iterations. But since we have overall 160 target tasks, so considering the randomness among tasks, we think it still gives meaningful result.

Moreover, remember that we have lower bound $\beta \epsilon_i^{-1}$ in our proposed algorithm, In our experiment for linear model, we actually find that only using a constant small number like 50 for each epoch is enough for getting meaningful result. While in convnet, considering the complexity of model, we still obey this law.

F.2. More results and analysis for linear model

F.2.1. MORE COMPREHENSIVE SUMMARY
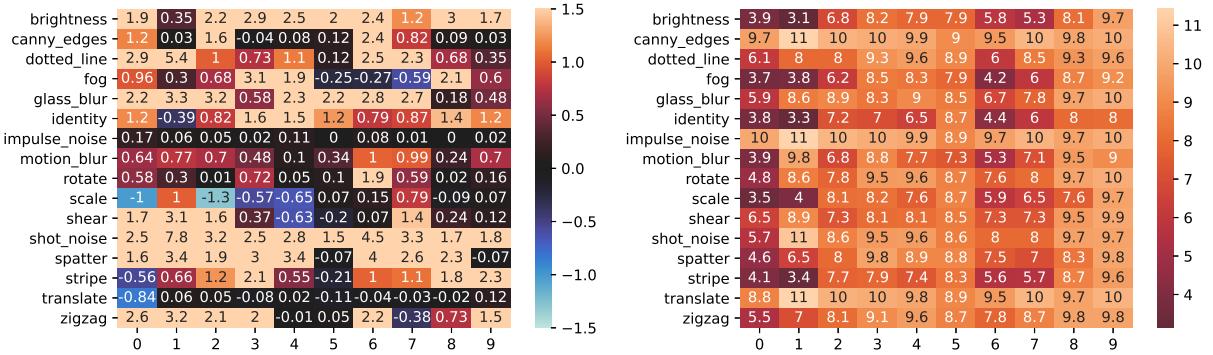
Figure 3. summary of performance difference for linear model (restated of figure 1); **left: The prediction difference (in %) between ada and non-ada for all target tasks right: The incorrect percentage of non-adaptive algorithm.Note that** 10% **is the baseline due to the in-balance dataset and the large the worse. Please refer to the main paper for further explanation**,

F.2.2. WHY OUR ALGORITHM FAILS ON SOME TARGET TASKS ?

Here we focus on why our algorithm get bad performance on some tasks. Overall, other than the randomness, this is mainly due to the incompatibility between our theoretical assumption (realizable linear model) and the complicated data structure in reality.

To be specific, there might a subset of source tasks that are informative about the target task under linear model assumptions, but other source tasks are far away from this model assumption. Due to the model misspecification, those misleading tasks may gain more sampling in the adaptive algorithm. We conjecture that this might be the case for target tasks like *scale_0* and *scale_2*. To further support our argument, we further analyze its sample number distribution and test error changing with increasing epoch in the next paragraph.

For the *scale_0* task, in Figure 8 we observe the non-stationary sample distribution changing across each epoch. But fortunately, the distribution is not extreme, there are still some significant sample from *X_0* source tasks. This aligns with the test error observation in Figure 9, which is still gradually decreasing, although slower than the non-adaptive algorithm. On the other hand, the sample distributions are even worse. We observed that nearly all the samples concentrate towards *X_5* source tasks. Thus no only we can not sample enough informative data, but we also force the model to fit to some non-related data. Such mispecifcation has been reflected in the test error changing plot. (You may notice the unstable error changing in non-adaptive algorithm performance, we think it is acceptable randomness because we only run each target task once and the target task itself is not very easy to learn.)
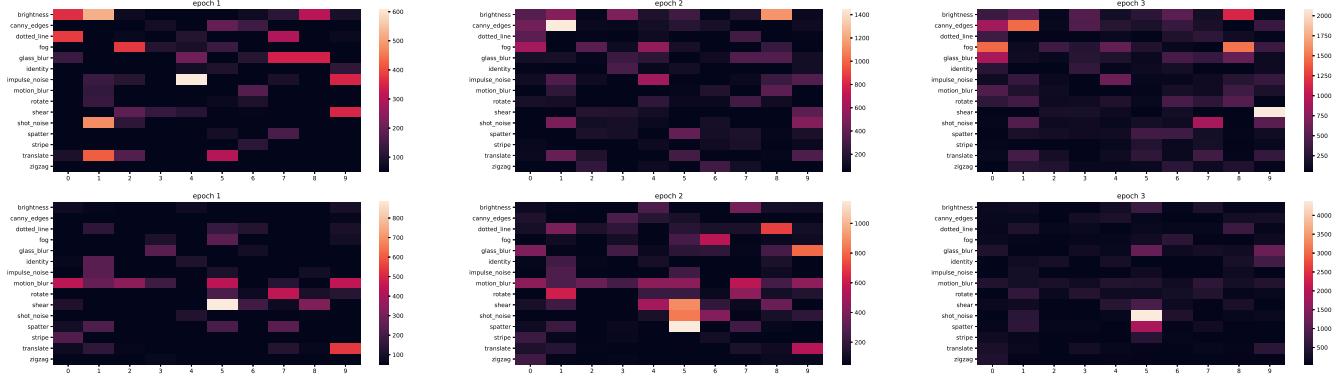
Figure 4. **top: sample distribution for target task as** *scale_0*, **bottom: sample distribution for target task as** *scale_2* We show the sample distribution at each epoch 1,2,3.
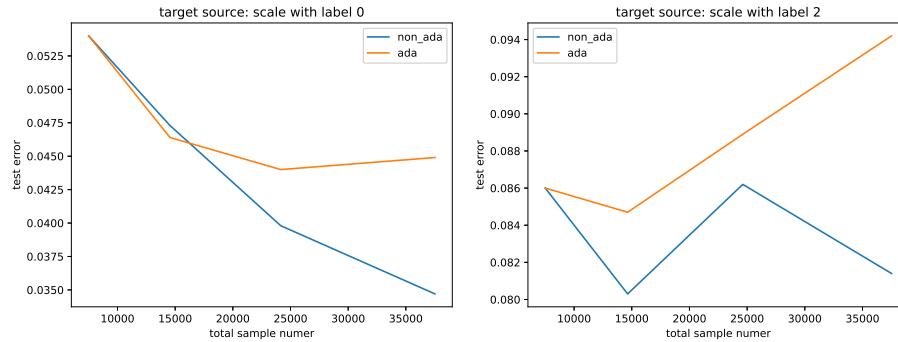

Figure 5. **Test error change after each epoch for target task as** *scale_0* **and** *scale_2*

F.2.3. MORE GOOD SAMPLE DISTRIBUTION EXAMPLES

Appendix F.3
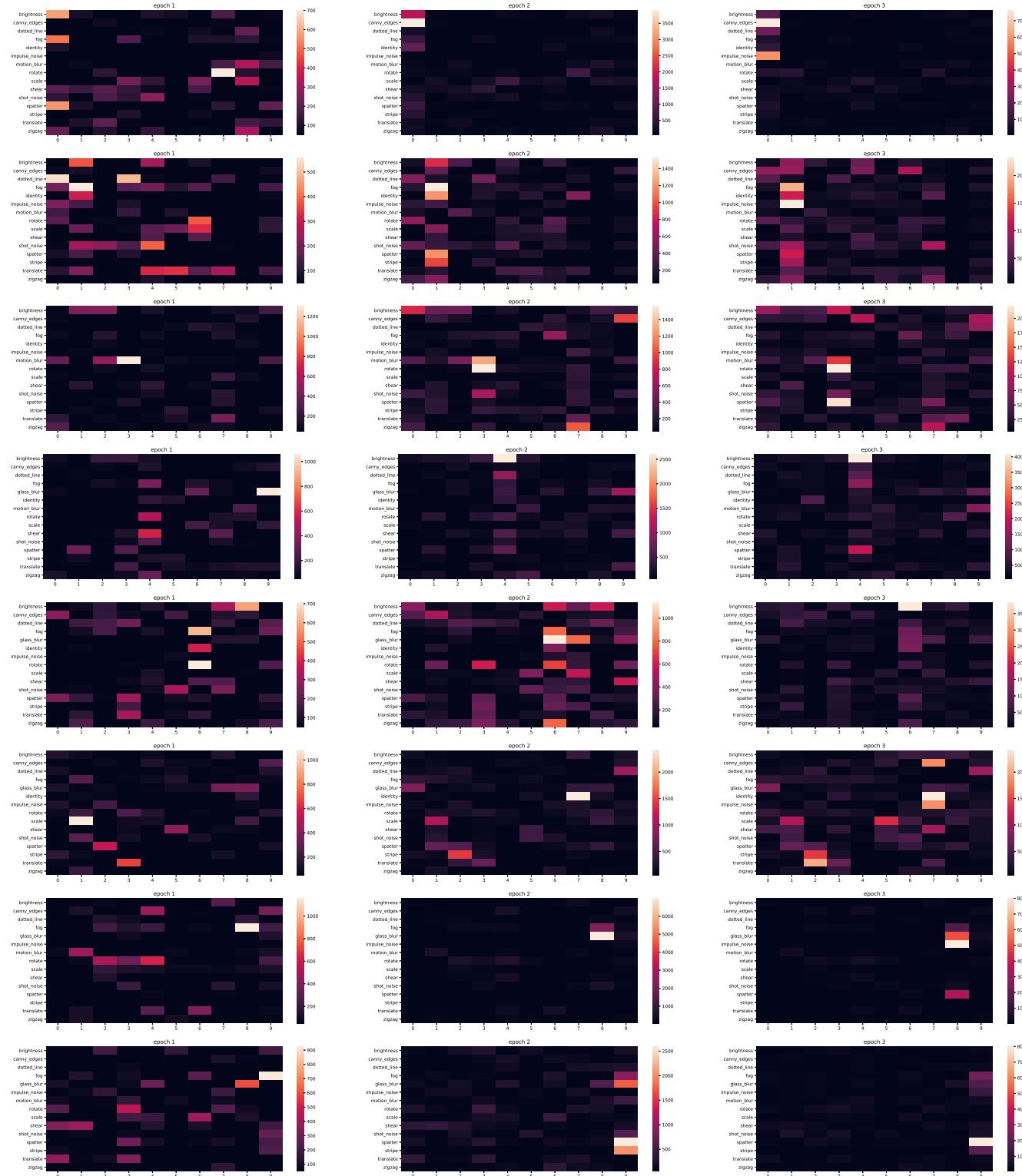
Figure 6. **Good sample distribution.** We show the sample distribution at each epoch 1,2,3. **From top to bottom:** *glass_blur_0,glass_blur_1,glass_blur_3,impulse_noise_4,motion_blur_6,motion_blur_7,identity_8,identity_9*

F.3. More results and analysis for convnet model

The convnet gives overall better accuracy than the linear model, except the translate class, as shown in Figure 7. So we want to argue that it might be harder for us to get as large improvement as on linear model given the better expressive power on convnet.
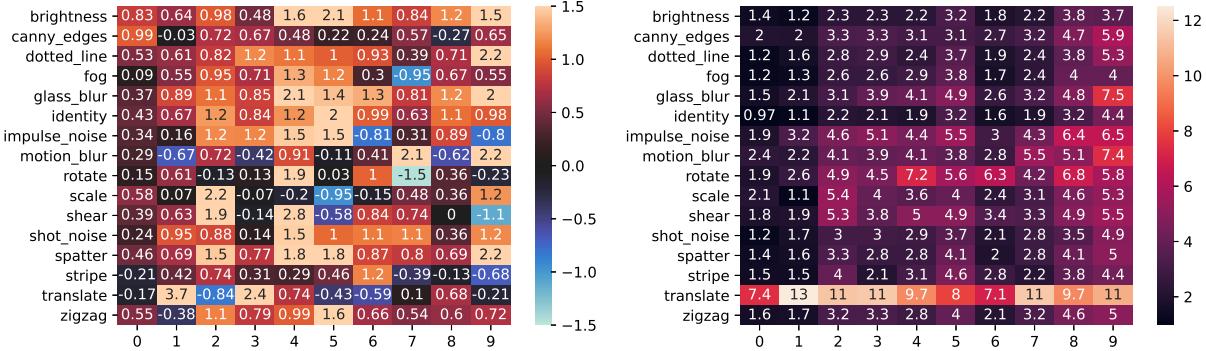
Figure 7. **left: summary of performance difference for conv model (restated of figure 2); right: the incorrect percentage of non-adaptive algorithm** On the right side we show the incorrect percentage of non-adaptive algorithm. Note that 10% is the baseline due to the in-balance dataset and the large the worse. Please refer to the main paper for further explanation,

F.3.1. WHY OUR ALGORITHM FAILS ON SOME TARGET TASKS ?

Here we show *scale_5* and *shear_9* as the representative bad cases. With the similar idea of linear model, we again observe the non-stationary sample distribution changing across each epoch in Figure 8. For *scale_5*, we observe that at the beginning of epoch 1, the sample fortunately converges to *X_5* source tasks, therefore our adaptive algorithm initially performs better than the non-adaptive one as shown in Figure 9. Unfortunately, the sample soon diverges to other source tasks, which more test error. For *shear_9*, although there are some samples concentrate on *X_9* source tasks, overall, the number of samples on *X_9* source tasks has a decrease proportion of total number of source sample. So the algorithm has a worse performance on this.
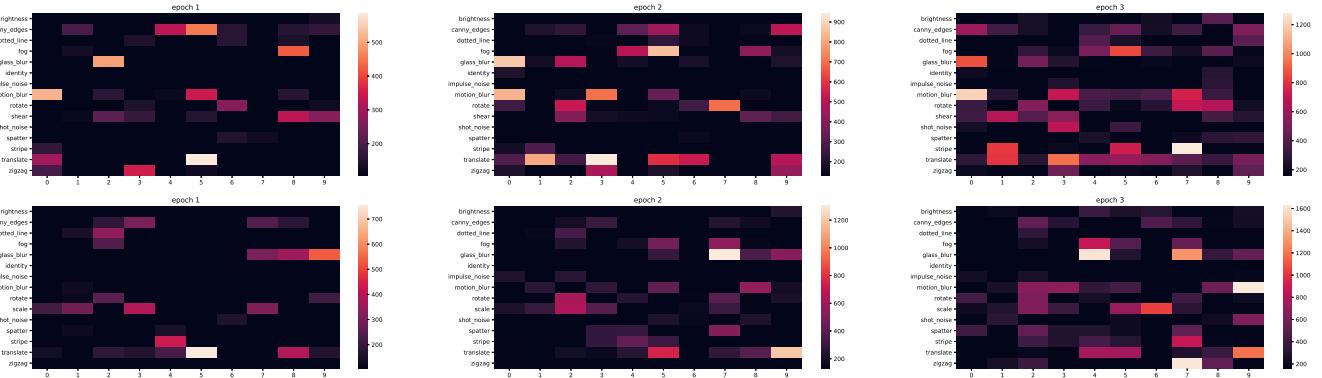

Figure 8. **top: sample distribution for target task as *scale_5*, bottom: sample distribution for target task as shear_9** We show the sample distribution at each epoch 1,2,3.
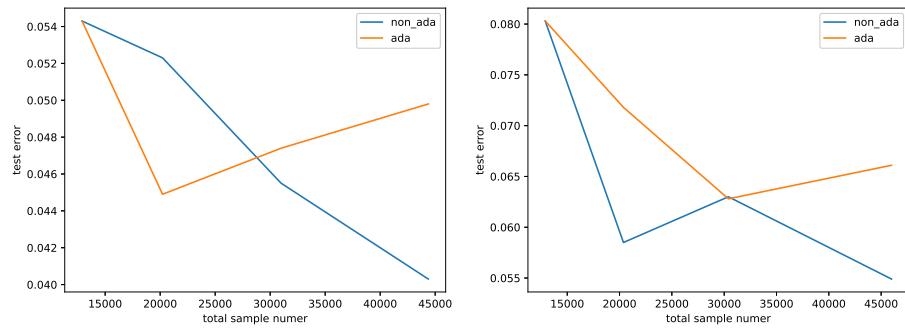
Figure 9. **Test error change after each epoch for target task as** *scale_5* **and** *shear_9*

F.3.2. MORE GOOD SAMPLE DISTRIBUTION EXAMPLES
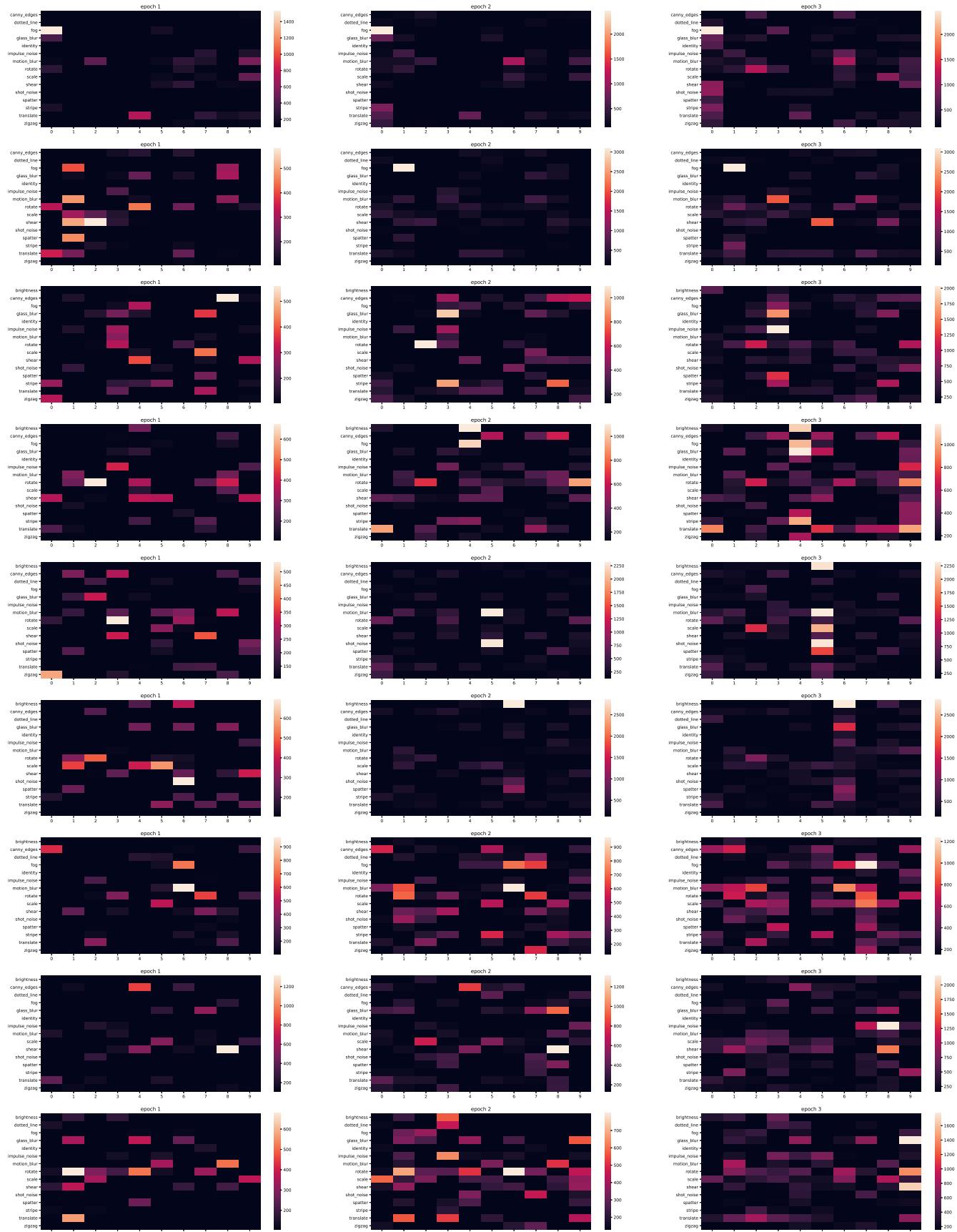
Figure 10. **Good sample distribution.** We show the sample distribution at each epoch 1,2,3. **From top to bottom:** *brightness_0,brightness_1,dotted_line_3,dotted_line_4,identity_5, fog_6,glass_blur_7,rotate_8,canny_edges_9*