Active Representation Learning for General Task Space with Applications in Robotics

Yifang Chen[1], **Yingbing Huang**[2], **Simon S. Du**[1*], **Kevin Jamieson**[1*], **Guanya Shi**[3*]

[1] Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle,WA
{yifangc, ssdu, jamieson, guanyas}@cs.washington.edu

[2] University of Illinois Urbana-Champaign, Champaign, IL
{yh21}@illinois.edu

[3] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA
{{guanyas }@andrew.cmu.edu
[*] Equal advising

Abstract

Representation learning based on multi-task pretraining has become a powerful approach in many domains. In particular, task-aware representation learning aims to learn an optimal representation for a specific target task by sampling data from a set of source tasks, while task-agnostic representation learning seeks to learn a universal representation for a class of tasks. In this paper, we propose a general and versatile algorithmic and theoretic framework for *active representation learning*, where the learner optimally chooses which source tasks to sample from. This framework, along with a tractable meta algorithm, allows most arbitrary target and source task spaces (from discrete to continuous), covers both task-aware and task-agnostic settings, and is compatible with deep representation learning practices. We provide several instantiations under this framework, from bilinear and feature-based nonlinear to general nonlinear cases. In the bilinear case, by leveraging the non-uniform spectrum of the task representation and the calibrated source-target relevance, we prove that the sample complexity to achieve ε-excess risk on target scales with $(k^*)^2\|v^*\|_2^2\varepsilon^{-2}$ where k^* is the effective dimension of the target and $\|v^*\|_2^2 \in (0, 1]$ represents the connection between source and target space. Compared to the passive one, this can save up to $\frac{1}{d_W}$ of sample complexity, where d_W is the task space dimension. Finally, we demonstrate different instantiations of our meta algorithm in synthetic datasets and robotics problems, from pendulum simulations to real-world drone flight datasets. On average, our algorithms outperform baselines by $20\% - 70\%$. [1]

1 Introduction

Recently, few-shot machine learning has enjoyed significant attention and has become increasingly critical due to its ability to derive meaningful insights for target tasks that have minimal data, a scenario commonly encountered in real-world applications. This issue is especially prevalent in robotics where data collection and training data is prohibitive to collect or even non-reproducible (e.g., drone flying with complex aerodynamics [1] or legged robots on challenging terrains [2]). One

[1] Code in https://github.com/cloudwaysX/ALMultiTask_Robotics

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

promising approach to leveraging the copious amount of data from a variety of other sources is multi-task learning, which is based on a key observation that different tasks may share a common low-dimensional representation. This process starts by pretraining a representation on source tasks and then fine-tuning the learned representation using a limited amount of target data ([3–7]).

In conventional supervised learning tasks, accessing a large amount of source data for multi-task representation learning may be easy, but processing and training on all that data can be costly. In real-world physical systems like robotics, this challenge is further amplified by two factors: (1) switching between different tasks or environments is often significantly more expensive (e.g., reset giant wind tunnels for drones [7]); (2) there are infinitely many environments to select from (i.e., environmental conditions are continuous physical parameters like wind speed). Therefore, it is crucial to minimize not only the number of samples, but the number of sampled source tasks, while still achieving the desired performance on the target task. Intuitively, not all source tasks are equally informative for learning a universally good representation or a target-specific representation. This is because source tasks can have a large degree of redundancy or be scarce in other parts of the task space. In line with this observation, Chen et al. [8] provided the first provable active representation learning method that improves training efficiency and reduces the cost of processing source data by prioritizing certain tasks during training with theoretical guarantees. On the other hand, many existing works [9–13] prove that it is statistically possible to learn a universally good representation by randomly sampling source tasks (i.e., the passive learning setting).

The previous theoretical work of [8] on active multi-task representation learning has three main limitations. First, it only focuses on a finite number of discrete tasks, treating each source independently, and therefore fails to leverage the connection between each task. This could be sub-optimal in many real-world systems like robotics for two reasons: (1) there are often infinitely many sources to sample from (e.g., wind speed for drones); (2) task spaces are often highly correlated (e.g., perturbing the wind speed will not drastically change the aerodynamics). In our paper, by considering a more general setting where tasks are parameterized in a vector space $\mathcal{W}$, we can more effectively leverage similarities between tasks compared to treating them as simply discrete and different. Secondly, the previous work only considers a single target, while we propose an algorithm that works for an arbitrary target space and distribution. This is particularly useful when the testing scenario is time-variant. Thirdly, we also consider the task-agnostic setting by selecting $\mathcal{O}(k)$ representative tasks among the d_W high dimension task space, where $k \ll d_W$ is the dimension of the shared representation. Although this result does not improve the total source sample complexity compared to the passive learning result in the bilinear setting [12], it reduces the number of tasks used in the training and therefore implicitly facilitates the training process.

In addition to those theoretical contributions, we extend our proposed algorithmic framework beyond a pure bilinear representation function, including the *known* nonlinear feature operator with unknown linear representation (e.g., random features with unknown coefficients), and the totally *unknown nonlinear representation* (e.g., deep neural network representation). While some prior works have considered nonlinear representations [9, 10, 14, 13] in passive learning, the studies in active learning are relatively limited [8]. All of these works only consider non-linearity regarding the input, rather than the task parameter. In this paper, we model task-parameter-wise non-linearity and show its effectiveness in experiments. Note that it particularly matters for task selections because the mapping from the representation space to task parameters to is no longer linear.

See more related works and how our problem scope is different from theirs in Appendix A.

1.1 Summery of contributions

- We propose the first generic active representation learning framework that admits any arbitrary source and target task space. This result greatly generalizes previous works where tasks lie in the discrete space and only a single target is allowed. To show its flexibility, we also provide discussions on how our framework can accommodate various supervised training oracles and optimal design oracles. (Section 3)
- We provide theoretical guarantees under a benign setting, where inputs are i.i.d. and a unit ball is contained in the overall task space, as a compliment to the previous work where tasks lie on the vertices of the whole space. In the target-aware setting, to identify an ε-good model our method requires a sample complexity of $\widetilde{\mathcal{O}}(kd_X(k^*)^2\|v^*\|_2^2 \min\{k^*, \kappa^2\}\varepsilon^{-2})$ where k^* is the effective dimension of the target, κ is the conditional number of representation matrix, and

$\|v^*\|_2^2 \in (0, 1]$ represents the connection between source and target space that will be specified in the main paper. Compared to passive learning, our result saves up to a factor of $\frac{k^2}{d_W}$ in the sample complexity when targets are uniformly spread over the k-dim space and up to $\frac{1}{d_W}$ when targets are highly concentrated. Our results further indicate the necessity of considering the continuous space by showing that directly applying the previous algorithm onto some discretized sources in the continuous space (e.g., orthonormal basis) can lead to worse result. Finally, ignoring the tasks used in the warm-up phases, in which only a few samples are required, both the target-aware and the target-agnostic cases can save up to $\widetilde{\mathcal{O}}(k^* + k)$ number of tasks compared to the passive one which usually requires d_W number of tasks. (Section 4)

- We provide comprehensive experimental results under different instantiations beyond the benign theoretical setting, studying synthetic and real-world scenarios: 1) For the synthetic data setting in a continuous space, we provide results for pure linear, known nonlinear feature operator ψ_X and unknown nonlinear representation ϕ_X. Our target-aware active learning (AL) approach shows up to a significant budget saving (up to 68%) compared to the passive approach and the target-agnostic AL approach also shows an advantage in the first two cases. 2) In a pendulum simulation with continuous task space, we provide the results for known nonlinear feature operator ψ_X and ψ_W and show that our target-aware AL approach has up to 20% loss reduction compared to the passive one, which also translates to better nonlinear control performance. 3) Finally, in the real-world drone dataset with a discrete task space, we provide results for unknown linear and nonlinear representation ϕ_X and show that our target-aware AL approach converges much faster than the passive one. (Section 5)

2 Preliminary

Multi-task (or multi-environments). Each task or environment is parameterized by a known vector $w \in \mathbb{R}^{d_W}$. We denote the source and target task parameter space as $\mathcal{W}_{\text{source}} \subset \mathbb{R}^{d_W}, \mathcal{W}_{\text{target}} \subset \mathbb{R}^{d_W}$. These spaces need not be the same (e.g., they could be different sub-spaces). In the discrete case, we set w as a one-hot encoded vector and therefore we have in total d_W number of candidate tasks while in the continuous space, there exist infinitely many tasks. For convenience, we also use w as the subscript to index certain tasks. In addition, we use $\nu_{\text{source}} \in \Delta(\mathcal{W}_{\text{source}}), \nu_{\text{target}} \in \Delta(\mathcal{W}_{\text{target}})$ to denote the task distribution for the sources and targets.

Data generation. Let $\mathcal{X} \in \mathbb{R}^{d_X}$ be the input space. We first assume there exists some *known* feature/augmentation operator $\psi_X : \mathcal{X} \to \mathbb{R}^{d_{\psi_X} \geq d_W}, \psi_W : \mathcal{W} \to \mathbb{R}^{d_{\psi_W} \geq d_W}$, that can be some nonlinear operator that lifts w, x to some higher dimensional space (e.g., random Fourier features [15]). Notice that the existence of non-identical ψ indicates the features are not pairwise independent and the design space of $\mathcal{W}_{\text{source}}$ is not benign (e.g., non-convex), which adds extra difficulty to this problem.

Then we assume there exists some *unknown* underlying representation function $\phi_X : \psi(\mathcal{X}) \to \mathcal{R}$ which maps the augmented input space $\psi(\mathcal{X})$ to a shared representation space $\mathcal{R} \in \mathbb{R}^k$ where $k \ll d_{\psi_X}, k \leq d_{\psi_W}$, and its task counterparts $\phi_W : \psi(\mathcal{W}) \to \mathcal{R}$ which maps parameterized task space to the feature space. Here the representation functions are restricted to be in some function classes Φ, e.g., linear functions, deep neural networks, etc.

In this paper, we further assume that ϕ_W is a linear function $B_W \in \mathbb{R}^{k \times d_{\psi_W}}$. To be more specific, for any fixed task w, we assume each sample $(x, y) \sim \nu_w$ satisfies

$$y = \phi_X(\psi_X(x))^\top B_W \psi_W(w) + \xi, \quad \xi \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

For convenience, we denote Z_w as the collection of n_w sampled data $(x_w^1, y_w^1), ..., (x_w^{n_w}, y_w^{n_w}) \sim \mu_w$. We note that when ψ_X, ψ_W is identity and ϕ_X is linear, this is reduced to standard linear setting in many previous papers [9, 11, 12, 8].

The task diversity assumption. There exists some distribution $p \in \Delta(\mathcal{W}_{\text{source}})$ that $\mathbb{E}_{w \sim p} \lambda_{\min}(B_W \psi_W(w) \psi_W(w)^\top B_W^\top) > 0$, which suggests the source tasks are diverse enough to learn the representation.

Data collection protocol. We assume there exists some i.i.d. data sampling oracle given the environment and the budget. To learn a proper representation, we are allowed access to an *unlimited* n_{source} number of data from source tasks during the learning process by using such an oracle. Then at

the end of the algorithm, we are given a few-shot of *mix* target data $Z_{\text{target}} = \{Z_w\}_{w\sim\nu_{\text{target}}}$ which is used for fine-tuning based on learned representation $\hat{\phi}_X$. Denote n_{target} as the number of data points in Z_{target}.

Data collection protocol for target-aware setting. When the target task is not a singleton, we additionally assume a few-shot of *known environment* target data $\dot{Z}_{\text{target}} := \{Z_w, w\}_{w\in\dot{W}_{\text{target}}}$, where $|\dot{W}_{\text{target}}| = \dim(\mathcal{W}_{\text{target}})$ and $\dot{W}_{\text{target}} = \{\arg\max_{W\in\mathcal{W}_{\text{target}}} \lambda_{\min}(WW^\top)\}$. Again denote $\dot{n}_{\text{target}}$ as the number of data points in $\dot{Z}_{\text{target}}$, we have $\dot{n}_{\text{target}} \approx n_{\text{target}}^{2/3} \ll n_{\text{source}}$.

Remark 2.1. Here $|\dot{W}_{\text{target}}|$ represents vectors that can cover every directions of $\mathcal{W}_{\text{target}}$ space. This extra $\dot{Z}_{\text{target}}$ requirement comes from the non-linearity of l_2 loss and the need to learn the relationship between sources and targets. We want to emphasize that such an assumption implicitly exists in previous active representation learning [8] since $\dot{Z}_{\text{target}} = Z_{\text{target}}$ in their single target setting. Nevertheless, in a passive learning setting, only mixed Z_{target} is required since no source selection process involves. Whether such a requirement is necessary for target-aware active learning remains an open problem.

Other notations. Let e_i to be one-hot vector with 1 at i-th coordinates and let $\epsilon_i = 2^{-i}$.

2.1 Goals

Expected excess risk. For any target task space $\mathcal{W}_{\text{target}}$ and its distribution ν_{target} over the space, as well as a few-shot examples as stated in section 2, our goal is to minimize the expected excess risk with our estimated $\hat{\phi}_X$

$$\text{ER}(\hat{\phi}_X, \nu_{\text{target}}) = \mathbb{E}_{w_0\sim\nu_{\text{target}}}\mathbb{E}_{(x,y)\sim\nu_{w_0}} \|\hat{\phi}_X(\psi_X(x))^\top \hat{w}_{\text{avg}} - y\|_2$$

where $\hat{w}_{\text{avg}} = \arg\min_w \sum_{(x,y)\in Z_{\text{target}}} \|\hat{\phi}_X(\psi_X(x))w - y\|_2$, which average model estimation that captures the data behavior under the expected target distribution. Note that the $\mathcal{W}_{\text{target}}, \nu_{\text{target}}$ are given in advance in the target-aware setting.

The number of tasks. Another side goal is to save the number of long-term tasks we are going to sample during the learning process. Since a uniform exploration over d_W^{source}-dimension is unavoidable during the warm-up stage, we define long-term task number as

$$\left|\left\{w \in \mathcal{W}_{\text{source}} \mid \ n_w \geq \tilde{\Omega}(\varepsilon^{-\alpha})\right\}\right|$$

where α is some arbitrary exponent and ε is the target accuracy and n_w is number of samples sampled from task w as defined above.

3 A general framework

Our algorithm 1 iteratively estimates the shared representation $\hat{\phi}_X, \hat{B}_W$ and the next target relevant source tasks which the learner should sample from by solving several optimal design oracles

$$g(f, A) = \min_{q\in\Delta(\mathcal{W}_{source})} \lambda_{\max}\left((\int q(w)f(w)f(w)^\top)^{-1}A\right) \tag{2}$$

This exploration and exploitation (target-aware exploration here) trade-off is inspired by the classical ϵ-greedy strategy, but the key difficulty in our work is to combine that with multi-task representation learning and different optimal design problems. The algorithm can be generally divided into three parts, and some parts can be skipped depending on the structure and the goal of the problem.

- **Coarse exploration:** The learner uniformly explores all the directions of the $\mathcal{W}_{\text{source}}$ (denoted by distribution q_0) in order to find an initial k-dimension subspace V that well spans over the representation space (i.e., $\frac{1}{c}B_W B_W^\top \leq B_W VV^\top B_W^\top \leq cB_W B_W^\top$ for some arbitrary constant $c \leq \frac{d_{\psi_W}}{k}$). To give an intuitive example, suppose $B_W \in \mathbb{R}^{2\times d_{\psi_W}^{\text{source}}+1}$ has the first half column equals e_1 and the second hard equals e_2. Then instead of uniformly choosing $\{e_i\}_{i\in[d_{\psi_W}^{\text{source}}]}$ task, we only need explore over two tasks $V[1] = \sqrt{\frac{2}{d_{\psi_W}^{\text{source}}}}[1, 1, \ldots, 0, 0, \ldots], V[2] = \sqrt{\frac{2}{d_W^{\text{source}}}}[0, 0, \ldots, 1, 1, \ldots]$.

Algorithm 1 Active multi-task representation learning (general templates)

Inputs: Candidate source set $\mathcal{W}_{\text{source}}$. Classes of candidate representation function Φ_X, Φ_W and the known feature operator ψ_X, ψ_W.
[Target-aware only] Inputs: Target set $\mathcal{W}_{\text{target}}$ and distribution ν_{target}. Few-shot sample $\dot{Z}_{\text{target}}$ as defined in the preliminary.
Stage 1: Coarse exploration. (Warm-up stage)
Set initial sampling distribution $q_0 = g(\psi_W, I_{d_{\psi_W}})$ where g is defined in Eqn. 2
Set $n_0 \approx \text{poly}(d_{\psi_X}, k) + \text{poly}(d_{\psi_W}, k)$. Collect $n_0 q_0(w)$ data for each task denoted as $\{Z_w\}_{w|q_0(w)\neq 0}$ and update $\hat{\phi}_X \leftarrow \mathcal{O}^X_{\text{offline }0}(\{Z_w\}_{w|q_0(w)\neq 0}, \psi_X)$ and $\hat{B}_W \leftarrow \mathcal{O}^W_{\text{offline}}(\{Z_w\}_{w|q_0(w)\neq 0}, \hat{\phi}_X)$
for $j = 1, 2, 3, \ldots$ **do**
  Stage 2: Fine target-agnostic exploration (Directly choose $q_1^j = q_0$ when $k = \Theta(d_W)$)
  Compute the exploration sampling distribution $q_1^j = g(\hat{B}_W \circ \psi_W, I_k)$
  $n_1^j \approx \text{poly}(d_{\psi_X}, k)\epsilon_j^{-\frac{4}{3}}$. Collect $n_1^j q_1^j(w)$ data for each task denoted as $\{Z_w\}_{w|q(w)\neq 0}$ and update $\hat{\phi}_X \leftarrow \mathcal{O}^X_{\text{offline }1}(\{Z_w\}_{w|q_1^j(w)\neq 0}, \psi_X)$ and $\hat{B}_W \leftarrow \mathcal{O}^W_{\text{offline}}(\{Z_w\}_{w|q_1^j(w)\neq 0}, \dot{Z}_{\text{target}}, \hat{\phi}_X)$
  [Target-aware only] Stage 3: Fine target-aware exploration
  Compute the exploitation sampling distribution $q_2^j = g(\hat{B}_W \circ \psi_W, \Sigma_{\text{regu}})$ where Σ_{regu} is the regularized version of $\hat{B}_W(\mathbb{E}_{w_0\sim\nu_0} w_0 w_0^\top)\hat{B}_W^\top$ after clipping out insignificant eigenvalues.
  Set $n_2^j \approx \text{poly}(d_{\psi_X}, k)\epsilon_j^{-2}$. Collect $n_2^j q_2^j(w)$ data for each task denoted as $\{Z_w\}_{w|q_2^j(w)\neq 0}$ and update $\hat{\phi}_X \leftarrow \mathcal{O}^X_{\text{offline }3}(\{Z_w\}_{w|q_1^j(w)\neq 0 \text{ and } q_2^j(w)\neq 0}, \psi_X)$.
end for
Return $\hat{\phi}_X$

We want to highlight that the sample complexity of this warm-up stage only scales with d_{ψ_X}, k and the spectrum-related parameters of B_W (i.e., $\kappa(B_W), \sigma_{\min}(B_X)$), not the desired accuracy ε.

- **Fine target-agnostic exploration:** The learner iteratively updates the estimation of V and uniformly explore for $\widetilde{\mathcal{O}}(\epsilon_j^{-\frac{4}{3}})$ times on this k, instead of d_{ψ_W} subspace, denoted by distribution q_1. (Note this $\epsilon_j^{-\frac{4}{3}}$ comes from the exploration part in ϵ-greedy, which is $(n_2^j)^{\frac{2}{3}}$) Such reduction not only saves the cost of maintaining a large amount of physical environment in real-world experiments but also simplifies the non-convex multi-task optimization problem. Of course, when $k = \Theta(d_{\psi_W})$, we can always uniformly explore the whole (d_{ψ_W} space as denoted in the algorithm. Note that theoretically, q_1 only needs to be computed once as shown in 4. In practice, to further improve the accuracy while saving the task number, the q_1 can be updated only when a significant change from the previous one happens, which is adopted in our experiments as shown in appendix E.1.
- **Fine target-aware exploration.** In the task-awareness setting, the learner estimates the most-target-related sources parameterized by $\{w\}$ based on the current representation estimation and allocates more budget on those, denoted by distribution q_2. By definition, q_2 should be more sparse than q_1 and thus allowing the final sample complexity only scales with k^*, which measures the effective dimension in the source space that is target-relevant.

Computational oracle for optimal design problem. Depending on the geometry of $\{\psi_W(w)\}_{w\in\mathcal{W}_{\text{source}}}$, the learner should choose proper offline optimal design algorithms to solve $g(f, A)$. Here we propose several common choices. 1). When $\mathcal{W}_{\text{source}}$ contains a ball, we can approximate the solution via an eigendecomposition-based closed-form solution with an efficient projection as detailed in Section 4. 2) When $\mathcal{W}_{\text{source}}$ is some other convex geometry, we can approximate the result via the Frank-Wolfe type algorithms [16], which avoids explicitly looping over the infinite task space. 3) For other even harder geometry, we can use discretization or adaptive sampling-based approximation [17]. In our experiments, we adopt the latter one and found out that its running time cost is almost neglectable in our pendulum simulator experiment in Section 5, where the ψ_W is a polynomial augmentation.

Offline optimization oracle $\mathcal{O}^X_{\text{offline}}$. Although we are in the continuous setting, the sampling distribution q_0, q_1, q_2 is sparse. Therefore, our algorithm allows any proper passive multi-task

learning algorithm, either theoretical or heuristic one, to plugin the $\mathcal{O}^X_{\text{offline}}$. Some common choices include gradient-based joint training approaches[18–21], the general non-convex ERM [9] and other more carefully designed algorithms [12, 22]. We implement the first one in our experiments (Section 5) to tackle the nonlinear ψ_X, ϕ_X and give more detailed descriptions of the latter two in Section 4 and Appendix B.1 to tackle the bilinear model.

4 A theoretical analysis under the benign $\mathcal{W}_{\text{source}}$ setting

4.1 Assumptions

Assumption 4.1 (Geometry of the task space)**.** *We assume the source task space $\mathcal{W}_{source}$ is a unit ball $\mathbb{B}^{d_W^{source}}(1)$ that span over the first $d_W^{source} \geq \frac{1}{2}d_W$ without loss of generality, while the target task space $\mathcal{W}_{target} \subset \mathbb{R}^{d_W}$ can be any arbitrary $\mathbb{B}^{d_W^{target}}(1)$.*

Under this assumption, we let B_W^{source} denote the first d_W^{source} columns of B_W, which stands for the source-related part of B_W. And B_W^{target}

Then we assume the bilinear model where $\phi_X = B_X \in \mathbb{B}^{d_X \times k}$ and $\psi_X, \psi_W = I$. Therefore, $d_{\psi_X} = d_X, d_{\psi_W} = d_W$. Moreover the model satisfies the following assumptions

Assumption 4.2 (Benign B_X, B_W)**.** *B_X is an orthonormal matrix. Each column of B_W has magnitude $\Theta(1)$ and $\sigma_{\min}(B_W^{source}) > 1$. Suppose we know $\bar{\kappa} \geq \kappa(B_W^{source}), \sigma_{\max}(B_W^{target})$ and $\underline{\sigma} \leq \sigma_{min}(B_W^{source}), \sigma_{min}(B_W^{target})$. Trivially, $\bar{\kappa} = \sqrt{d_W}, \underline{\sigma} = 1$.*

Finally, the following assumption is required since we are using a training algorithm in [12] and might be able to relax to sub-gaussian by using other suboptimal oracles.

Assumption 4.3 (Isotropic Gaussian Input)**.** *For each task w, its input i satisfies $x_{i,w} \sim \mathcal{N}(0, I_d)$.*

4.2 Algorithm

Here we provide the target-aware theory and postpone the target-agnostic in the Appendix. C since its analysis is covered by the target-aware setting.

This target-aware algorithm 2 follows the 3-stage which corresponds to sampling distribution q_0, q_1, q_2 with explicit solutions. Notice that calculating q_1 once is enough for theoretical guarantees.

We use existing passive multi-task training algorithms as oracles for $\mathcal{O}^X_{\text{offline 1}}, \mathcal{O}^X_{\text{offline 2}}$ and use the simple ERM methods for $\mathcal{O}^W_{\text{offline}}$ based on the learned $\hat{B}$. For the coarse exploration and fine target-agnostic exploration stage, the main purpose is to have a universal good estimation in all directions of B_X. (i.e., upper bound the $\sin(\hat{B}_X, B_X)$) Therefore we choose the alternating minimization (MLLAM) proposed in [12]. On the other hand, for the fine target-aware exploration, we mainly care about final transfer learning performance on learned representation. Therefore, we use a non-convex ERM from [9]. We defer the details and its theoretical guarantees for $\mathcal{O}_{\text{offline}}$ into Appendix B.1.

Note the major disadvantage from [9] comes from its sample complexity scaling with a number of training source tasks, which will not be a problem here since in $\mathcal{O}^X_{\text{offline, 3}}$ since only $k + k^* \ll d_W$ number of tasks are used. The major benefit of using non-convex ERM comes from its generality that it works even for the non-linear setting and is not tied with a specific algorithm. That is to say, as long as there exists other theoretical or heuristic oracles $\mathcal{O}^X_{\text{offline, 1}}, \mathcal{O}^X_{\text{offline, 2}}$ giving a similar guarantee, stage 3 always works.

4.3 Results

Theorem 4.1 (Informal)**.** *By running Algo. 2, in order to let $ER(\hat{\phi}_X, \nu_{target}) \leq \varepsilon^2$ with probability $1 - \delta$, the number of source samples n_{source} is at most*

$$\widetilde{\mathcal{O}}\left((kd_X + \log(1/\delta))\,(k^*)^2 \min\{k^*, \kappa^2(B_W)\} \max_i \|W_i^*\|_2^2 \varepsilon^{-2} + \textit{low-order}\right)$$

Here $k^* = rank(\mathbb{E}_{w_0 \sim \nu_{target}} B_W w_0 w_0^\top B_W^\top)$ represents the effective dimension of target and

$$W_i^* = \underset{w \in \mathcal{W}_{source}}{\arg\min} \|w\|_2 \quad s.t \quad B_W^{source} w = u_i \sqrt{\lambda_i} \text{ where } U, \Lambda \leftarrow Eig(\mathbb{E}_{w_0 \sim \nu_{target}} B_W w_0 w_0^\top B_W^\top).$$

Algorithm 2 Target-aware algorithm for benign source space

Inputs: Target probability $\delta, \bar{\kappa}, \underline{\sigma}$. Some constant $\beta_1, \beta_2, \beta_3$. Others same as Algo. 1.

Set q_0 as $q_0(e_t) = \frac{1}{d_W}, \forall t \in d_W$, and$q_0(w) = 0$ otherwise

Set $n_0 = \beta_1 \bar{\kappa}^2 \left(k^3 d_X \bar{\kappa}^2 + d_W^{\frac{3}{2}} \underline{\sigma}^{-2} \sqrt{k + \log(1/\delta)} \right)$. Collect $n_0 q_0(w)$ data for each task denoted as $\{Z_w\}_{w|q(w)\neq 0}$

Update $\hat{B}_X \leftarrow \mathcal{O}^X_{\text{offline 1}}(\{Z_w\}_{w|q_0(w)\neq 0})$ and $\hat{B}_W^{\text{source}} \leftarrow \mathcal{O}^W_{\text{offline}}(\{Z_w\}_{w|q_0(w)\neq 0}, \hat{B}_X)$

Compute q_1 as $q_1(v_i) = \frac{1}{k}, \forall i \in k$, and $q_0(w) = 0$ otherwise. Here v_i is the i-th vector of V, where $U, D, V \leftarrow \text{SVD}(\hat{B}_W^{\text{source}})$

for $j = 1, 2, 3, \ldots$ **do**

  Set $n_1^j = \beta_2 \epsilon_j^{-\frac{4}{3}} k^{\frac{5}{3}} d_W^{\frac{2}{3}} d_X^{\frac{1}{3}} \left(k^{\frac{2}{3}} d_W^{\frac{1}{3}} \underline{\sigma}^{-\frac{4}{3}} + \bar{\kappa}^2 \underline{\sigma}^{-\frac{2}{3}} \right)$. Collect $n_1^j q_1(w)$ data for each task denoted as $\{Z_w\}_{w|q_1(w)\neq 0}$.

  Update $\hat{B}_X \leftarrow \mathcal{O}^X_{\text{offline 2}}(\{Z_w\}_{w|q_1(w)\neq 0})$, $\hat{B}_W^{\text{source}} \leftarrow \mathcal{O}^W_{\text{offline}}(\{Z_w\}_{w|q_1(w)\neq 0}, \hat{B}_X)$ and $\hat{B}_W^{\text{target}} \leftarrow \mathcal{O}^W_{\text{offline}}(\dot{Z}_{\text{target}}, \hat{B}_X)$

  Find a set of target-aware tasks parameterized by $\tilde{W}_j$ with each column i as

$$\tilde{W}_j(i) = \text{Proj}_{\mathcal{W}_{\text{source}}} w_i' = \frac{w_i'}{\|w_i'\|_2}$$

$$\text{where } w_i' = \arg\min_w \|w\|_2 \quad \text{s.t.} \quad \hat{B}_{W,j}^{\text{source}} w = u_i \sqrt{\lambda_i} \quad \forall \Lambda_i \geq 8(k d_W)^{\frac{3}{2}} \sqrt{\frac{d_X}{n_1}}$$

$$\text{where } U, \Lambda \leftarrow \text{Eig}\left(\mathbb{E}_{w_0 \sim \nu_{\text{target}}} \left[\hat{B}_{W,j}^{\text{target}} w_0 (\hat{B}_{W,j}^{\text{target}} w_0)^\top \right] \right)$$

  Compute q_2^j as $q_2^j(w) = \frac{1}{\#\,\text{col}(\tilde{W}_j)}, \forall w \in \text{col}(\tilde{W}_j)$ and $q_2^j(w) = 0$ otherwise

  Assign n_2^j total sampling budget as $\#\,\text{col}(\tilde{W}_j) \beta_3 \max_i \|W_j'(i)\|_2^2 \epsilon_j^{-2}$

  Collect $n_2^j(w) = n_2^j q_2^j(w)$ data for each task denoted as $\{Z_w\}_{w|q_2(w)\neq 0}$.

  Update the model, note that both data collected from stage 2 and stage 3 are used.

$$\tilde{B}_X \leftarrow \mathcal{O}^X_{\text{offline 3}}(\{Z_w\}_{w|q_1(w)\neq 0 \text{ and } q_2(w)\neq 0})$$

end for

Return $\tilde{B}_X$

As long as the number of target samples satisfies

$$n_{target} \geq \tilde{\Omega}((k + \log(1/\delta))\varepsilon^{-2}), \quad \dot{n}_{target} \gtrapprox \widetilde{\Omega}\left(\varepsilon^{-\frac{4}{3}} (k^*)^{\frac{2}{3}} \sqrt{k} \left(d_W^{\frac{1}{2}} \underline{\sigma}^{-\frac{4}{3}} + k^{-\frac{2}{3}} d_W^{\frac{1}{6}} \bar{\kappa}^2 \underline{\sigma}^{-\frac{1}{3}} \right) \right)$$

Comparison with passive learning. By choosing $\{e_i\}_{i \in [d_W^{\text{source}}]}$ as a fixed source set, we reduce the problem to a discrete setting and compare it with the passive learning. In [9], the authors get N_{total} as most $\frac{k d_X d_W \|\mathbb{E}_{w_0 \sim \nu_{\text{target}}} B_W w_0 w_0^\top B_W^\top\|}{\sigma^2_{\min}(B_W^{\text{source}})} \varepsilon^{-2}$. We first consider the cases in their paper that the target task is uniformly spread $\|\mathbb{E}_{w_0 \sim \nu_{\text{target}}} B_W w_0 w_0^\top B_W^\top\| = \frac{1}{k}$.

- When the task representation is well-conditioned $\sigma^2_{\min}(B_W^{\text{source}}) = \frac{d_W}{k}$. We have a passive one as $\widetilde{\mathcal{O}}(k d_X \varepsilon^{-2})$ while the active one $\widetilde{\mathcal{O}}(k d_X \frac{k^2}{d_W} \varepsilon^{-2})$ (See Lemma B.8 for details), which suggests as long as $d_W \gg k^2$, our active learning algorithm gain advantage even in a relatively uniform spread data and representation conditions.
- Otherwise, we consider the extreme case that $\sigma^2_{\min}(B_W^{\text{source}}) = 1$. We have passive one $\widetilde{\mathcal{O}}(d_X d_W \varepsilon^{-2})$ while the active one $\widetilde{\mathcal{O}}(k^3 d_X \varepsilon^{-2})$. Notice here we require $d_W \gg k^3$.

Both of them indicate the necessity of considering the continuous case with large d_W even if everything is uniformly spread. On the other hand, whether we can achieve the same result as the passive one when $d_W \leq k^3$ remains to be explored in the future.

We then consider the single target w_0 case.

- With well-conditioned B_W, the passive one now has sample complexity $\mathcal{O}(k^2 d_X \varepsilon^{-2})$ while the active gives a strictly improvement $\mathcal{O}(\frac{k^2 d_X}{d_W} \varepsilon^{-2})$.
- With ill-conditioned B_W where $\sigma_{\min}(B_W) = 1$ and $\max_i \|W_i^*\| = 1$, that is, only a particular direction in source space contributes to the target. The Passive one now has sample complexity $\mathcal{O}(k d_X d_W \varepsilon^{-2})$ while our active one only has $k d_X \varepsilon^{-2}$, which demonstrates the benefits of our algorithm in unevenly distributed source space.

Comparison with previous active learning. By using the same discrete reduction and set single target w_0, we compare our result with the current state-of-art active representation algorithm in [23]. They achieves $\widetilde{\mathcal{O}}(k d_X \|\nu\|_1^2 \varepsilon^{-2})$, where $\nu = \arg\min_\nu \|\nu\|_1$ s.t $B_W \nu = B_W w_0$. On the other hand, our active one gives $\widetilde{\mathcal{O}}(k d_X \|w^*\|_2^2 \varepsilon^{-2})$, where $w^* = \arg\min_\nu \|\nu\|_2$ s.t $B_W \nu = B_W w_0$, which is strictly better than the discrete one. This again indicates the separation between continuous and discrete cases where in fixed discrete sets, the L_1 norm regularization is strictly better than L_2.

Furthermore, when a fixed discrete set is given, which is exactly the setting in [23]. Their algorithm can be seen as a computationally efficient reduction under ours.(Appendix B.5.)

Save task number. When ignoring the short-term initial warm-up stage, we only require maintaining $\widetilde{\mathcal{O}}(k + \log(N_{\text{total}} k^*))$ number of source tasks, where the first term comes from q_1 in the target-agnostic stage and the second term comes from q_2 in the target-aware stage.

5 Experiment

In this section, we provide experimental results under different instantiations of the Algorithm 1, and all of them show the effectiveness of our strategy both in target-aware and target-agnostic settings.

5.1 Settings

Datasets and problem definition. Our results cover the different combinations of ψ_X, ϕ_X, ψ_W as shown in Table 1. Here we provide a brief introduction for the three datasets and postpone the details into Appendix E. [2]

	identity ψ_W	nonlinear ψ_W
identity ψ_X and linear ϕ_X	synthetic, drone	NA
nonlinear ψ_X and linear ϕ_X	synthetic	pendulum simulator
identity ψ_X and nonlinear ϕ_X	synthetic, drone	NA

Table 1: Summary of different instantiations

- **Synthetic data.** We generate data that strictly adhere to our data-generating assumptions and use the same architecture for learning and predicting. When ϕ_X is nonlinear, we use a neural network ϕ_X to generate data and use a slightly larger neural net for learning. The goal for synthetic data is to better illustrate our algorithm as well as serve as the first step to extend our algorithm on various existing datasets.
- **Pendulum simulator.** To demonstrate our algorithm in the continuous space. we adopt the multi-environment pendulum model in [24] and the goal is to learn a w-dependent residual dynamics model $f(x, w) \in \mathbb{R}$ where x is the pendulum state and $w \in \mathbb{R}^5$ including external wind, gravity and damping coefficients. $f(x, w)$ is highly nonlinear with respective to x and w. Therefore we use known non-linear feature operators ψ_X, ψ_W. In other words, this setting can be regarded as a misspecified linear model. It is also worth noting that due to the non-invertibility of ψ_W, the explicit selection of a source via a closed form is challenging. Instead, we resort to an adaptive sampling-based method discussed in Section 3. Specifically, we uniformly sample w from the source space, select the best w', and then uniformly sample around this w' at a finer grain. Our findings indicate that about 5 iterations are sufficient to approximate the most relevant source.
- **Real-world drone flight dataset [7].** The Neural-Fly dataset [7] includes real flight trajectories using two different drones in various wind conditions. The objective is to learn the residual

[2] Github Link: https://github.com/cloudwaysX/ALMultiTask_Robotics

aerodynamics model $f(x, w) \in \mathbb{R}^3$ where $x \in \mathbb{R}^{11}$ is the drone state (including velocity, attitude, and motor speed) and w is the environment condition (including drone types and wind conditions). We collect 6 different w and treat each dimension of $f(x, w)$ as a separate task. Therefore w is reformulated as a one-hot encoded vector in $\mathbb{R}^{18}$.

For each dataset/problem, we can choose different targets. For simplicity, in the following subsection, we present results for one target task for each problem with 10 random seeds regarding random data generation and training, and put more results in Appendix E. In all the experiments, we use a gradient-descent joint training oracle, which is a standard approach in representation learning.

5.2 Results

Those results encapsulate the effectiveness of active learning in terms of budget utilization and test loss reduction. In the drone dataset, we further demonstrate its ability in identifying relevant source tasks (see Figure 2). We note that in two robotics problems (pendulum simulation and real-world drone dataset), the active learning objective is to learn *a better dynamics model*. However, in the pendulum simulation, we deploy a model-based nonlinear controller which translates better dynamics modeling to enhanced control performance (see Figure 1 and Appendix E.2).

	Target-aware AL	Target-agnostic AL
identity ψ_X and linear ϕ_X	38.7%	51.6%
nonlinear ψ_X and linear ϕ_X	38.7%	45.2%
identity ψ_X and non-linear ϕ_X	32.0%	68.0%

Table 2: Results on synthetic data. Using the test loss of the final output model from passive learning as a baseline, we show the ratio between the budget required by target-aware/target-agnostic active learning to achieve a similar loss and the budget required by passive learning.

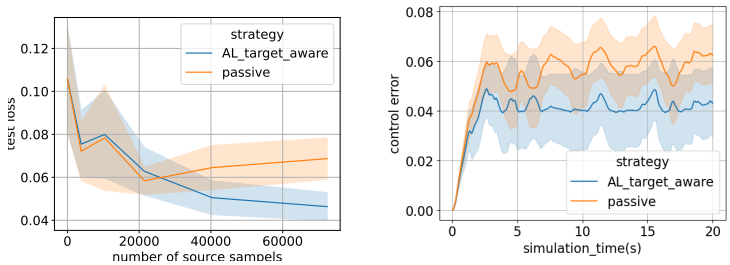

Figure 1: Results on pendulum simulator for a specific target. **Left:** The test loss of the estimated model $\hat{f}$. The passive strategy suffers from negative transfer while the active strategy steadily decreases. **Right:** The control error using final output $\hat{f}$. Here we use a model-based nonlinear policy $\pi(x, \hat{f})$. The model learned from active strategy leads to better control performance.

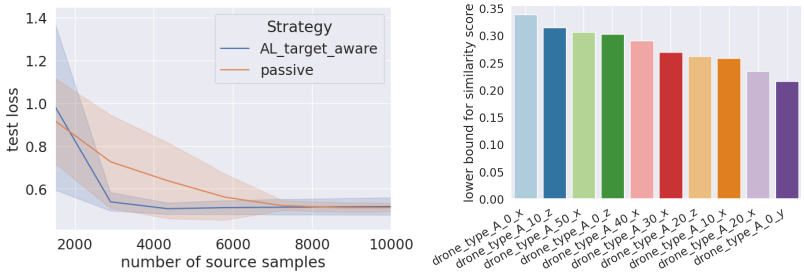

Figure 2: Results on the real drone dataset [7] with target `drone_type_A_30_z`. Source data includes two drone types A and B, six wind speeds from 0 to 50, and three directions x-y-z. We present results for linear ϕ_X here and postpone the non-linear ϕ_X case in Appendix E.3. **Left:** The test loss of the estimated bilinear model $\hat{f}$. The passive strategy converges slower than the active strategy. **Right:** Top 10 the most similar source tasks. Given the target environment, the algorithm successfully finds the other `drone_type_A` environments as relevant sources. See more explanations in Appendix E.3.

References

[1] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE, 2019.

[2] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[5] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[6] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736, 2022.

[7] Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66):eabm6597, 2022.

[8] Yifang Chen, Kevin Jamieson, and Simon Du. Active multi-task representation learning. In *International Conference on Machine Learning*, pages 3271–3298. PMLR, 2022.

[9] Simon S. Du, Wei Hu, Sham M. Kakade, Jason D. Lee, and Qi Lei. Few-shot learning via learning the representation, provably, 2021.

[10] Nilesh Tripuraneni, Michael I. Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity, 2020.

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

[12] Kiran Koshy Thekumparampil, Prateek Jain, Praneeth Netrapalli, and Sewoong Oh. Sample efficient linear meta-learning by alternating minimization. *arXiv preprint arXiv:2105.08306*, 2021.

[13] Ziping Xu and Ambuj Tewari. Representation learning beyond linear prediction functions. *Advances in Neural Information Processing Systems*, 34:4792–4804, 2021.

[14] Liam Collins, Aryan Mokhtari, Sewoong Oh, and Sanjay Shakkottai. Maml and anil provably learn representations. In *International Conference on Machine Learning*, pages 4238–4310. PMLR, 2022.

[15] Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *2008 46th annual allerton conference on communication, control, and computing*, pages 555–561. IEEE, 2008.

[16] Michael J Todd. *Minimum-volume ellipsoids: Theory and algorithms*. SIAM, 2016.

[17] Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. Theoretical foundation for cma-es from information geometry perspective. *Algorithmica*, 64:698–716, 2012.

[18] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

[19] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

[20] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

[21] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44 (9):5149–5169, 2021.

[22] Shuxiao Chen, Koby Crammer, Hangfeng He, Dan Roth, and Weijie J. Su. Weighted training for cross-task learning, 2021.

[23] Yiping Wang, Yifang Chen, Kevin Jamieson, and Simon Shaolei Du. Improved active multi-task representation learning via lasso. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35548–35578. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/wang23b.html`.

[24] Guanya Shi, Kamyar Azizzadenesheli, Michael O'Connell, Soon-Jo Chung, and Yisong Yue. Meta-adaptive nonlinear control: Theory and algorithms. *Advances in Neural Information Processing Systems*, 34:10013–10025, 2021.

[25] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020.

[26] Rahul Ramesh and Pratik Chaudhari. Model zoo: A growing" brain" that learns continually. *arXiv preprint arXiv:2106.03027*, 2021.

[27] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021.

[28] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[29] Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. *arXiv preprint arXiv:1905.12917*, 2019.

[30] Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Task-robust model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, 33:18860–18871, 2020.

[31] Xingcheng Yao, Yanan Zheng, Xiaocong Yang, and Zhilin Yang. Nlp from scratch without large-scale pretraining: A simple and efficient framework. In *International Conference on Machine Learning*, pages 25438–25451. PMLR, 2022.

[32] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. *arXiv preprint arXiv:2302.03169*, 2023.

[33] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023.

Contents

1 Introduction **1**

1.1 Summery of contributions 2

2 Preliminary **3**

2.1 Goals 4

3 A general framework **4**

4 A theoretical analysis under the benign $\mathcal{W}_{\text{source}}$ setting **6**

4.1 Assumptions 6

4.2 Algorithm 6

4.3 Results 6

5 Experiment **8**

5.1 Settings 8

5.2 Results 9

A Related works **14**

B Result and analysis for target-aware **14**

B.1 Offline training oracles used in Algorithm 14

B.1.1 Choice of $\mathcal{O}^{X}_{\text{offline 1}}$ 14

B.1.2 Choice of $\mathcal{O}^{X}_{\text{offline 2}}$ 15

B.1.3 Choice of $\mathcal{O}^{W}_{\text{offline}}$ 15

B.2 Excess risk analysis 15

B.2.1 Reduce to an optimal design problem 16

B.2.2 Bound decomposition and the excess risk result 17

B.2.3 Detail proofs for warm-up stage 18

B.2.4 Detail proofs for task-agnostic exploration strategy 19

B.2.5 Auxillary lemmas 20

B.3 Lemmas about the properties of W' 22

B.4 Sample complexity analysis – Formal version of Theorem 4.1 24

B.5 Algorithms in [23] is a special case of Algo. 2 25

B.6 More interpretation on results 25

C Results and analysis for target-agnostic **26**

C.1 Algorithm for target-agnostic 26

C.2 Results and analysis 26

C.3 Compare to previous passive learning and the target-aware one 27

D Limitations from the theoretical perspective **27**

E Experiment details **28**

E.1 Synthetic data 28

E.1.1 Settings 28

E.1.2 Results 30

E.2 Pendulum simulator 32

E.2.1 Settings 32

E.2.2 Results 33

E.3 Real-world drone flight dataset 33

E.3.1 Settings 33

E.3.2 Results 33

A Related works

Here we give a brief summary of other representation learning or multi-task papers that are related but different in some aspects

Multi-task with negative correlation Some multi-task works [25–27, 22] assume different tasks don't share the same representation, so learning on one task may hurt another. They usually group similar tasks and assign an independent model to each group [25–27] or assign high weights on target-relevant sources [22]. The essential difference between those work and ours is that they assume a pass over the whole dataset is possible and aim to achieve the ultimate best performance, whereas we assume it is not (setting a large amount of experiment environment or maintaining a long time real data collection is costly). Consequently, they should not be considered as active.

Passive Multi-task training/Meta learning While our paper focuses on data collection, some papers focus on the training process with some given dataset. For example, [22] mentioned above reweighting and joint-training all tasks. Another large topic in this scope is called "Meta-learning" [28–30], which usually focuses on more detailed updating methods. In conclusion, this line of works is parallel to our work, and all those methods can be regarded as a plug-in oracle in 1 Line 5, 9, 12.

Sample-wise data selection for representation learning Classical pool-based active learning selects most informative data for a single task. Recently, some works [31–33] started to focus on selecting helpful data from a large corpus of web-scale for some known target task, where web-scale data could be seen as a mix of multi-task data without explicit "task" information. Besides, those works usually focus on coarse labels and self-contrastive learning. Therefore, although they also aim to learn a presentation/pretrained model from non-target data, their detailed settings are quite different from ours.

B Result and analysis for target-aware

B.1 Offline training oracles used in Algorithm

B.1.1 Choice of $\mathcal{O}^X_{\text{offline 1}}$

To better illustrate this oracle $\mathcal{O}^X_{\text{offline}}$, we first give the following definition.

Definition B.1 (Modified from Assumption 2 in [12])**.** *For any t tasks with parameter matrix $\dot{V} = [\dot{v}_1, \dot{v}_2, \dots, \dot{v}_t] \in \mathbb{R}^{d_W \times t}$. Let λ_1^* and λ_k^* denote the largest and smallest eigenvalues of the task diversity matrix $(k/t) B_W^{source} \dot{V} \dot{V}^\top (B_W^{source})^\top \in \mathbb{R}^{k \times k}$ respectively. Then we say $\dot{V}$ is μ-incoherent, i.e.,*

$$\max_{i \in [t]} \|B_W^{source} \dot{v}_i\|^2 \le \mu \lambda_k^*$$

Notice that here $\dot{V}$ is a general representation of collected source tasks used for training in the different stages. Therefore, the λ_k^*, μ is also defined differently corresponding to each stage. Specially, we have

- **Stage 1(data collected by q_0)**:
  - $t = d_W, \dot{V} = I_{d_W}$
  - $\lambda_k^* = \frac{k}{d_W} \sigma_k^2(B_W^{\text{source}})$
  - $\mu \ge \frac{1}{\lambda_k^*}$
- **Stage 2(data collected by q_1)**:
  - $t = k, \dot{V} = V$ where $\_, \_, V \leftarrow \text{SVD}(\hat{B}_W^{\text{source}})$ as defined in line 5
  - $\lambda_k^* = \sigma_k^2(B_W^{\text{source}})$
  - $\mu \ge \frac{\sigma_{\max}^2(B_W^{\text{source}})}{\lambda_k^*}$

Note that $\lambda_k^* = \sigma_k^2(B_W^{\text{source}})$ in the stage 2 comes from $B_W^{\text{source}} \dot{V} \dot{V}^\top (B_W^{\text{source}})^\top = \Theta(B_W^{\text{source}} (B_W^{\text{source}})^\top)$ which will be proved later. Therefore, applying these results to

Now we restate the generalization guarantees from a fixed design (passive learning)

Theorem B.1 (Restate Theorem 1 in [12])**.** *Let there be t linear regression tasks, each with m samples, and*

$$m \geq \widetilde{\Omega}\left(\left(1 + k\left(\sigma/\sqrt{\lambda_k^*}\right)^2\right) k \log t + k^2\right), \quad \text{and } mt \geq \widetilde{\Omega}\left(\left(1 + \left(\sigma/\sqrt{\lambda_k^*}\right)^2\right)(\lambda_1^*/\lambda_k^*)\,\mu d_X k^2\right)$$

Then MLLAM, initialized at $\hat{B}_X = U_{\text{init}}$ s.t. $\left\|\left(\mathbf{I} - B_X (B_X)^\top\right) U_{\text{init}}\right\|_F \leq \min\left(3/4, O\left(\sqrt{\lambda_k^*/\lambda_1^*}\right)\right)$ and run for $K = \lceil \log_2\left(\lambda_k^* \lambda_k^* mt/\lambda_1^* \sigma^2 \mu d_X k^2\right) \rceil$ iterations, outputs $\hat{B}_X$ so that the following holds $(w.p.\ \geq 1 - K/(d_X k)^{10})$

$$\sin(\hat{B}_X, B_X) \leq \left\|\left(\mathbf{I} - B_X (B_X)^\top\right) \hat{B}_X\right\|_F \leq \widetilde{O}\left(\left(\frac{\sigma}{\sqrt{\lambda_k^*}}\right) k \sqrt{\frac{\mu d_X}{mt}}\right)$$

Specifically, suppose we satisfy all the requirements in the theorem and run the proper amount of times, then we can guarantee $\hat{B}_X$ after each stage j with w.h.p $\geq 1 - 2K/(d_X k)^{10}$

- **Stage 1(data collected by q_0)**: $\sin(\hat{B}_X, B_X) \leq \widetilde{\mathcal{O}}\left(\sigma k \sqrt{\frac{d_X}{n_0}}\right)$
- **Stage 2(data collected by q_1)**: $\sin(\hat{B}_X, B_X) \leq \widetilde{\mathcal{O}}\left(\sigma k \sqrt{\frac{d_X \sigma_{\max}^2(B_W^{\text{source}})}{n_0}}\right)$

Let Event $\mathcal{E}_{\text{offline 1}}$ denote the above guarantees hold for all epochs.

B.1.2 Choice of $\mathcal{O}^X_{\text{offline 2}}$

We use the ERM from [9]. For readers' convenience, we restate the formal definition of oracle below

$$\hat{B}_X = \arg\min_B \sum_{w | q_1(w) \neq 0 \text{ and } q_2(w) \neq 0} \arg\min_w \sum_{(x,y) \in Z_w} \|x^\top w - y\|_2$$

By using this ERM with the follow-up finetune on Z_{target}, we get the following claims. Note that this claim comes from some part of Proof of Theorem 4.1 in the previous paper and has also been used in Claim 3 in [8].

Claim B.1. *By running the ERM-based algorithm, we get the following upper bounds,*

$$\text{ER}(\tilde{B}_X, \nu_{target}) \leq \mathbb{E}_{w_0 \sim \nu_{target}}\left[\frac{\|P^\perp_{X_{target}\hat{B}_X} X_{target} B_X B_W w_0\|^2}{n_{target}} + \sigma^2 \frac{k + \log(1/\delta)}{n_{target}}\right]$$

We need to admit that, from a theoretical perspective, we choose this oracle since we can directly use their conclusions. But other oracles like $\mathcal{O}^X_{\text{offline 2}}$ might also work.

B.1.3 Choice of $\mathcal{O}^W_{\text{offline}}$

This is the ERM oracle based on learned $\hat{B}_X$. Specially, we have $\hat{B}_W^{\text{source/target}} \leftarrow \mathcal{O}^W_{\text{offline}}(\{Z_w\}_{w|q(w)\neq 0}, \hat{B}_X)$ defined as

$$\hat{B}_W^{\text{source/target}} = \sum_{w|q(w)\neq 0} \hat{w}_w w^\top, \text{ where } \hat{w}_w = \arg\min_{\hat{w} \in \mathbb{R}^k} \sum_{(x,y)\sim Z_w} \|x^\top \hat{B}_X^\top \hat{w} - y\|_2,$$

B.2 Excess risk analysis

Theorem B.2 (Excess risk guarantees)**.** *By running the Algo. 2, after epoch j, as long as $\mathcal{E}_{\text{offline 1}}$ holds, we have w.h.p $1 - \delta$,*

$$\text{ER}(\tilde{B}_X, \nu_{target}) \leq \widetilde{\mathcal{O}}\left(\sigma^2 k d_X k^* \epsilon_j^2\right)$$

as long as

$$\dot{n}_{target} \geq \epsilon_j^{-\frac{4}{3}} d_X^{-\frac{2}{3}} \left(k^{-\frac{2}{3}} d_W^{\frac{1}{2}} \underline{\sigma}^{-\frac{4}{3}} + k^{-\frac{4}{3}} d_W^{\frac{1}{6}} \bar{\kappa}^2 \underline{\sigma}^{-\frac{1}{3}} \right) \sqrt{k + \log(d_W/\delta)}$$
$$n_{target} \geq \epsilon_j^{-2} d_X^{-1} (k^*)^{-1} \frac{k}{k + \log(d_W/\delta)}$$

Proof. Here we provide the proof sketches, which will be specified in the following sections.

In Section B.2.1, we first reduce $\mathrm{ER}(\tilde{B}_X, \nu_{\text{target}})$ to an optimal design problem by showing that, with a proper number of n_{target},

$$\mathrm{ER}(\tilde{B}_X, \nu_{\text{target}}) \lessapprox (kd_X + \log(1/\delta)) \mathrm{Tr} \left(\left((B_W^{\text{source}}) \left(\sum_{w \in \mathcal{S}} n_w w w^\top \right) (B_W^{\text{source}})^\top \right)^{-1} B_W \left(\mathbb{E}_{\nu_{\text{target}}} w w^T \right) B_W^\top \right)$$

It is easy to see that, as long as B_W is known. The problem is reduced to an optimal design problem with fixed optimization target.

So the main challenge here is to iteratively estimate B_X, B_W and design the budget allocation to different sources. Therefore, in Section B.2.2, we further decompose the it into

$$\mathrm{Tr} \left(\left((B_W^{\text{source}}) \left(\sum_{w \in \mathcal{S}} n_w w w^\top \right) (B_W^{\text{source}})^\top \right)^{-1} B_W \left(\mathbb{E}_{\nu_{\text{target}}} w w^T \right) B_W^\top \right)$$
$$\leq \underbrace{\mathbb{E}_{w_0 \sim \nu_{\text{target}}} \left(\left[(B_W w_0)^\top \square B_W w_0 \right] - \mathrm{Tr} \left(\beta_3 (B_W W')^\top \square B_W W' \right) \right.}_{\text{target agnostic exploration error}} + \beta_3 \underbrace{\mathrm{Tr} \left((B_W W')^\top \square B_W W' \right)}_{\text{target-aware exploration error}}$$

where $\square = \left(B_W \left(\sum_{w \in \mathcal{S}} n_w w w^\top \right) B_W^\top \right)^{-1}$. Here the **target-aware exploration error** captures the error from selecting the target-related sources (defined by q_2). On the other hand, the **target agnostic exploration error** captures the error from model estimation and the uniform exploration.

Now the main challenge here is to upper-bound the model estimation error. Specifically, the estimation comes from Coarse exploration (Stage 1) and Fine target-agnostic exploration (Stage 2). Specifically, in Section B.2.3, we show that the k-dim-subspace represented by q_1 is a good course approximation up to *multiplicative* error. Then in Section B.2.4, we further tight the upper bound using data collected according to up to some *additive* error. □

B.2.1 Reduce to an optimal design problem

For any fixed epoch j, let n_w^j denotes the samples collected so far for task w and $\mathcal{S}$ denotes the set of tasks used in computing $\tilde{B}_X$. Therefore, we have $\mathcal{S} = \{w | q_1(w) \neq 0 \text{ and } \tilde{q}(w) \neq 0\}$ and $n_w \geq n_2(w) + n_2^j(w)$. For convenience, we omit the superscript j in the rest of the proofs.

From Claim B.1, it is easy to see that our main target is to optimize $\mathbb{E}_{w_0 \sim \nu_{\text{target}}} \| P^\perp_{X_{\text{target}} \hat{B}_X} X_{\text{target}} B_X B_W w_0 \|^2$. Decompose $B_W \left(\sum_{w \in \mathcal{S}} n_w w w^\top \right) B_W^\top$ as $UDU^\top$ and let $\Sigma_W = U \sqrt{D} U^\top$. As long as Σ_W is full rank, which we will prove later in Section B.2.3, we

have with probability $1-\delta$,

$$
\begin{aligned}
&\mathbb{E}_{w_0\sim\nu_{\text{target}}}\|P^{\perp}_{X_{\text{target}}\hat{B}_X}X_{\text{target}}B_XB_Ww_0\|^2\\
&=\mathbb{E}_{w_0\sim\nu_{\text{target}}}\|P^{\perp}_{X_{\text{target}}\hat{B}_X}X_{\text{target}}B_X\Sigma_W^{\frac{1}{2}}\Sigma_W^{-\frac{1}{2}}B_Ww_0\|^2\\
&\le\mathbb{E}_{w_0\sim\nu_{\text{target}}}\|P^{\perp}_{X_{\text{target}}\hat{B}_X}X_{\text{target}}B_X\Sigma_W^{\frac{1}{2}}\|_F^2\|\Sigma_W^{-\frac{1}{2}}B_Ww_0\|^2\\
&=\mathbb{E}_{w_0\sim\nu_{\text{target}}}\|P^{\perp}_{X_{\text{target}}\hat{B}_X}X_{\text{target}}B_XB_W\tilde{W}_{\mathcal{S}}\|_F^2(B_Ww_0)^\top\left(B_W\left(\sum_{w\in\mathcal{S}}n_www^\top\right)B_W^\top\right)^{-1}B_Ww_0\\
&=\|P^{\perp}_{X_{\text{target}}\hat{B}_X}X_{\text{target}}B_XB_W\tilde{W}_{\mathcal{S}}\|_F^2\text{Tr}\left(\left(B_W\left(\sum_{w\in\mathcal{S}}n_www^\top\right)B_W^\top\right)^{-1}B_W\left(\mathbb{E}_{\nu_{\text{target}}\in\Delta(\mathcal{W}_{\text{target}})}ww^T\right)B_W^\top\right)\\
&\lessapprox\sigma^2n_{\text{target}}\left(kd_X+\log(1/\delta)\right)\text{Tr}\left(\left(B_W\left(\sum_{w\in\mathcal{S}}n_www^\top\right)B_W^\top\right)^{-1}B_W\left(\mathbb{E}_{\nu_{\text{target}}}ww^T\right)B_W^\top\right)\\
&=\sigma^2n_{\text{target}}\left(kd_X+\log(1/\delta)\right)\text{Tr}\left(\left((B_W^{\text{source}})\left(\sum_{w\in\mathcal{S}}n_www^\top\right)(B_W^{\text{source}})^\top\right)^{-1}B_W\left(\mathbb{E}_{\nu_{\text{target}}}ww^T\right)B_W^\top\right)
\end{aligned}
$$

Therefore, we aim to minimize the $\text{Tr}\left(\left((B_W^{\text{source}})\left(\sum_{w\in\mathcal{S}}n_www^\top\right)(B_W^{\text{source}})^\top\right)^{-1}B_W\left(\mathbb{E}_{\nu_{\text{target}}}ww^T\right)B_W^\top\right)$. As we mentioned before, this is a pure optimal design problem if B_W is known in advance.

B.2.2 Bound decomposition and the excess risk result

Let $\square=\left(B_W\left(\sum_{w\in\mathcal{S}}n_www^\top\right)B_W^\top\right)^{-1}$, we have

$$
\begin{aligned}
&\mathbb{E}_{w_0\sim\nu_{\text{target}}}\left[(B_Ww_0)^\top\square B_Ww_0\right]\\
&=\underbrace{\mathbb{E}_{w_0\sim\nu_{\text{target}}}(\left[(B_Ww_0)^\top\square B_Ww_0\right]-\text{Tr}\left(\beta_3(B_WW')^\top\square B_WW'\right)}_{\text{target agnostic exploration error}}+\beta_3\underbrace{\text{Tr}\left((B_WW')^\top\square B_WW'\right)}_{\text{target-aware exploration error}}
\end{aligned}
$$

We first deal with the target-aware exploration error. It is easy to see that

$$
\begin{aligned}
&\beta_3\text{Tr}\left((B_WW')^\top\square B_WW'\right)\\
&=\beta_3\text{Tr}\left(\left(B_W\sum_w q_1(w)n_1ww^\top(B_W)^\top+B_W\sum_w q_2(w)n_2ww^\top(B_W)^\top\right)^{-1}B_WW'(B_WW')^\top\right)\\
&\le\text{Tr}\left(\left(\max_i\|\tilde{W}(i)\|_{2(\infty)}^2B_W\epsilon^{-2}\tilde{W}\tilde{W}^\top(B_W)^\top\right)^{-1}B_WW'(B_WW')^\top\right)\\
&\le\text{Tr}\left(\left(B_W\epsilon^{-2}W'(W')^\top(B_W)^\top\right)^{-1}B_WW'(B_WW')^\top\right)\\
&=\epsilon^2\text{rank}(\hat{B}_WW'(W')^\top\hat{B}_W^\top)\\
&\le\epsilon^2\text{rank}(B_W\mathbb{E}_{\nu_{\text{target}}}[w_0w_0^\top]B_W^\top)
\end{aligned}
$$

where the last equality comes from Lemma B.5.

We then deal with the target-agnostic exploration term. Let the clipping threshold in Line 9 be $\bar{\gamma}_j$. That is, ignoring all $\lambda_i \leq \bar{\gamma}$. Now, for $\beta_3 \geq 8$, when event $\mathcal{E}_{\text{offline 1}}$, holds, we have w.h.p $1 - d_W\delta$

$$
\begin{aligned}
&\mathbb{E}_{w_0\sim\nu_0}\left[(B_W w_0)^\top \square B_W w_0\right] - \beta_3 \text{Tr}\left((B_W W')^\top \square B_W W'\right) \\
&= \mathbb{E}_{w_0\sim\nu_0}\text{Tr}\left(\square\left(B_W w_0 (B_W w_0)^\top - 4\hat{B}_W^{\text{target}} w_0 (\hat{B}_W^{\text{target}} w_0)^\top\right)\right) \\
&\quad + \mathbb{E}_{w_0\sim\nu_0}\text{Tr}\left(\square\left(\frac{1}{2}\beta_3 \hat{B}_W^{\text{source}} w'(\hat{B}_W w')^\top - \beta_3 B_W^{\text{source}} w'(B_W w')^\top\right)\right) \\
&\quad + \mathbb{E}_{w_0\sim\nu_0}\text{Tr}\left(\square\left(4\hat{B}_W^{\text{target}} w_0 (\hat{B}_W^{\text{target}} w_0)^\top - \frac{1}{2}\beta_3 \hat{B}_W^{\text{source}} W'(\hat{B}_W^{\text{source}} W')^\top\right)\right) \\
&\leq \mathbb{E}_{w_0\sim\nu_0}\text{Tr}\left(\left(4B_W^{\text{target}} w_0 (B_W^{\text{target}} w_0)^\top - 4\hat{B}_W^{\text{target}} w_0 (\hat{B}_W^{\text{target}} w_0)^\top\right)\right)\|\square\| \\
&\quad + \beta_3 \text{Tr}\left(\left(\frac{1}{2}\hat{B}_W^{\text{source}} w'(\hat{B}_W w')^\top - \frac{1}{2}\dot{B}_W W'(\dot{B}_W W')^\top\right)\right)\|\square\| \\
&\quad + k\bar{\gamma}\|\square\| \\
&\leq \|\square\|\|\mathbb{E}[w_0 w_0^T]\|_*\|(B_W^{\text{target}})^\top B_W^{\text{target}} - (\hat{B}_W^{\text{target}})^\top \hat{B}_W^{\text{target}}\| + \|\square\|\|W'(W')^\top\|_*\|\dot{B}_W^\top \dot{B}_W - \hat{B}_W^\top \hat{B}_W\| + k\bar{\gamma}\|\square\| \\
&\leq 2\|\square\|\|\mathbb{E}[w_0 w_0^T]\|_*\|B_W^{\text{target}} - \hat{B}_W^{\text{target}}\|\|B_W^{\text{target}}\| \\
&\quad + 2\|\square\|\|\|W'(W')^\top\|_*\|\dot{B}_W - \hat{B}_W^{\text{source}}\|\|\hat{B}_W^{\text{source}}\| \\
&\quad + k\bar{\gamma}\|\square\| \\
&\leq \epsilon^2
\end{aligned}
$$

where the second two terms in the first inequality come from Section B.2.3 and the last term in the first inequality comes from the definition of W'. Here $\dot{B}_W = B_W V V^\top = B_W^{\text{source}} V V^\top$ is a pseudo representation of B_W^{source}, where V is the one calculated in Line 5. And the last inequality comes from the results in Section B.2.4. Notice that the probability $1 - d_W\delta$ comes from the union bound on all the calls of $\mathcal{O}_{\text{offline}}^W$.

Now combine the bounds above, we have

$$
\text{ER}(\tilde{B}_X, \nu_{\text{target}}) \leq \sigma^2\left(k d_X \log((\kappa N_i)/d_W) + \log\frac{1}{\delta}\right)k^*\epsilon^2
$$

B.2.3 Detail proofs for warm-up stage

After the first stage, according to Section B.1.1, as long as $\mathcal{E}_{\text{offline 1}}$ holds, we have

$$
\sin(\hat{B}_X, B_X) \leq \widetilde{\mathcal{O}}\left(\sigma k\sqrt{\frac{d_X}{n_0}}\right)
$$

Therefore, by Lemma B.2, we have with probability $1 - d_W\delta$,

$$
\begin{aligned}
\|\hat{B}_W^{\text{source}} - B_W^{\text{source}}\| &\leq 2\sqrt{k}\sin(\hat{B}_X, B_X)\|B_W\| + \sqrt{\frac{d_W}{n_0}}(k + \log(2/\delta)^{\frac{1}{4}} d_W^{\frac{1}{4}} \\
&\leq 2k^{\frac{3}{2}}\sqrt{\frac{d_X}{n_0}}\|B_W\| + 2d_W^{\frac{3}{4}}(k + \log(2/\delta)^{\frac{1}{4}}\sqrt{\frac{1}{n_0}}
\end{aligned}
$$

As long as $n_0 \geq 1024\bar{\kappa}^2\left(k^3 d_X \bar{\kappa}^2 + \frac{d_W^{\frac{3}{2}}}{\underline{\sigma}^2}\sqrt{k + \log(1/\delta)}\right)$, by using the Lemma B.1 below, we have for any arbitrary matrix M,

$$
\frac{1}{2}B_W M (B_W)^\top \leq \dot{B}_W M \dot{B}_W^\top \leq \frac{3}{2}B_W M (B_W)^\top
$$

In the other word, $\dot{B}$ can be regarded as a pseudo representation of B_W^{source}. In all the later epochs, when exploring k-subspace according to q_1^j, the learner actually learns $\dot{B}_W$.

Lemma B.1 (Guarantee on exploration basis 1)**.** *Suppose we have the estimated $\hat{B}_W$ satisfies*

$$8\|B_W - \hat{B}_W\|\|B_W\| \le \frac{1}{2}\lambda_{\min}(B_W B_W^\top)$$
$$\dot{V} \leftarrow \textit{column space of } \mathrm{SVD}(\hat{B}_W),$$

then let $\dot{B}_W = B_W \dot{V}\dot{V}^\top$, we have, for any arbitrary matrix M,

$$\frac{1}{2}B_W M (B_W)^\top \le \dot{B}_W M \dot{B}_W^\top \le \frac{3}{2} B_W M (B_W)^\top$$

Proof.

$$\begin{aligned}
&\dot{B}_W M \dot{B}_W^\top - B_W M (B_W)^\top \\
&= \dot{B}_W M \dot{B}_W^\top - \hat{B}_W M (\hat{B}_W)^\top + \hat{B}_W M (\hat{B}_W)^\top - B_W M B_W^\top \\
&= (\dot{B}_W - \hat{B}_W) M B_W^\top + \hat{B}_W M (\dot{B}_W - \hat{B}_W)^\top + (\hat{B}_W - B_W) M (\hat{B}_W)^\top + B_W M (\hat{B}_W - B_W)^\top \\
&= (B_W - \hat{B}_W)\dot{V}\dot{V}^\top M B_W^\top + \hat{B}_W M \dot{V}\dot{V}^\top (B_W - \hat{B}_W)^\top + (\hat{B}_W - B_W) M (\hat{B}_W)^\top + B_W M (\hat{B}_W - B_W)^\top
\end{aligned}$$

Therefore, according to our assumption, we can upper bound the above as

$$\begin{aligned}
\dot{B}_W M \dot{B}_W^\top - B_W M (B_W)^\top &\le 2\|B_W - \hat{B}_W\| \left(\|\hat{B}_W\| + \|B_W\|\right) M \\
&\le \left(4\|B_W - \hat{B}_W\|\|B_W\| + 2\|B_W - \hat{B}_W\|_2^2\right) M \\
&\le 8\|B_W - \hat{B}_W\|\|B_W\| M \\
&\le \frac{1}{2}\lambda_{\min}(B_W B_W^\top) M \le \frac{1}{2} B_W M B_W^\top
\end{aligned}$$

Similarly, it can be lower bounded by $-\frac{1}{2}B_W M B_W^\top$. Therefore we can get the target result by rearranging. □

B.2.4 Detail proofs for task-agnostic exploration strategy

First, we upper bound two $\|B_W - \hat{B}_W\|$ terms. From section B.1.1, as long as $\mathcal{E}_{\text{offline 1}}$ holds, we have

$$\sin(\hat{B}_X, B_X) \le \widetilde{\mathcal{O}}\left(k\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{source}}\|\right)$$

Therefore, by Lemma B.2, we have w.h.p at least $1 - (k + d_W^{\text{target}})\delta$

$$\begin{aligned}
\|\hat{B}_W^{\text{source}} - B_W^{\text{source}}\| &\le 2\sqrt{k}\sin(\hat{B}_X, B_X)\|B_W^{\text{source}}\| + \sqrt{\frac{k}{n_1}}(k + \log(2/\delta)^{\frac{1}{4}} k^{\frac{1}{4}} \\
&\le 2k^{\frac{3}{2}}\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{source}}\|^2 \\
\|\hat{B}_W^{\text{target}} - B_W^{\text{target}}\| &\le 2k\sin(\hat{B}_X, B_X)\|B_W^{\text{target}}\| + \sqrt{\frac{1}{\dot{n}_{\text{target}}}}(k + \log(2/\delta)^{\frac{1}{4}} (d_W^{\text{target}})^{\frac{1}{4}} \\
&\le 2k^{\frac{3}{2}}\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{target}}\|^2 + 2\sqrt{\frac{1}{\dot{n}_{\text{target}}}}(k + \log(2/\delta)^{\frac{1}{4}} (d_W^{\text{target}})^{\frac{1}{4}} \\
&\le 4k^{\frac{3}{2}}\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{target}}\|^2
\end{aligned}$$

where the last equality holds as long as $\dot{n}_{\text{target}} \ge n_1 \frac{\sqrt{(k+\log(2/\delta))d_W^{\text{target}}}}{k^3 d_X \|B_W^{\text{target}}\|^2}$.

Next, we upper bound the $\|W'(W')\|$ according to Lemma B.7.

$$\begin{aligned}\|W'(W')^\top\|_* &\lessapprox \frac{1}{\sigma^2_{\min}(B_W^{\text{source}})}\|B_W^{\text{target}}\mathbb{E}_{\nu_{\text{target}}}[w_0 w_0^\top](B_W^{\text{target}})^\top\|_* \\ &\leq \frac{1}{\sigma^2_{\min}(B_W^{\text{source}})}\|B_W^{\text{target}}\|^2\|\mathbb{E}_{\nu_{\text{target}}}[w_0 w_0^\top]\|_* \\ &\leq \frac{1}{\sigma^2_{\min}(B_W^{\text{source}})}\|B_W^{\text{target}}\|^2\end{aligned}$$

Finally, we have, by definition

$$\begin{aligned}\bar{\gamma} &\leq 2\|\hat{B}_W^{\text{target}} - B_W^{\text{target}}\|\|B_W^{\text{target}}\| \\ \|\square\| &\leq \frac{k}{n_1\sigma^2_{\min}(\dot{B}_W)} \lessapprox \frac{k}{n_1\sigma^2_{\min}(B_W^{\text{source}})}\end{aligned}$$

Combine all above, we have the upper bound

$$\begin{aligned}&\|\square\|\left(\|\mathbb{E}[w_0 w_0^T]\|_*\|B_W^{\text{target}} - \hat{B}_W^{\text{target}}\|\|B_W^{\text{target}}\| + \|\|W'(W')^\top\|_*\|\dot{B}_W - \hat{B}_W^{\text{source}}\|\|B_W^{\text{source}}\| + k\bar{\gamma}\right) \\ &\lessapprox \frac{k}{n_1\sigma^2_{\min}(B_W^{\text{source}})} * k^{\frac{3}{2}}\sqrt{\frac{d_X}{n_1}} * \left(k\|B_W^{\text{target}}\|^3 + \|B_W^{\text{source}}\|\|B_W^{\text{target}}\|^2\bar{\kappa}^2\right) \\ &\leq k^{\frac{5}{2}}d_X^{\frac{1}{2}}n_1^{-\frac{3}{2}}\|B_W^{\text{target}}\|^2\left(\frac{k\|B_W^{\text{target}}\|}{\underline{\sigma}^2} + \frac{\bar{\kappa}^3}{\underline{\sigma}}\right) \\ &\leq k^{\frac{5}{2}}d_X^{\frac{1}{2}}n_1^{-\frac{3}{2}}d_W\left(k\sqrt{d_W}\underline{\sigma}^{-2} + \bar{\kappa}^3\underline{\sigma}^{-1}\right)\end{aligned}$$

As long as $n_1 \geq \epsilon_j^{-\frac{4}{3}}k^{\frac{5}{3}}d_W^{\frac{2}{3}}d_X^{\frac{1}{3}}\left(k^{\frac{2}{3}}d_W^{\frac{1}{3}}\underline{\sigma}^{-\frac{4}{3}} + \bar{\kappa}^2\underline{\sigma}^{-\frac{2}{3}}\right)$, we have the final bound ϵ_j^2.

B.2.5 Auxillary lemmas

Lemma B.2. *Consider any t regression tasks parameterized by $\{\dot{v}_i\}_{i\in[n]}$. Denote $\dot{V} = [\dot{v}_1, \dot{v}_2, \ldots, \dot{v}_t]$ and $|X_{\dot{v}_i}| = n$ for all $i \in [t]$, define*

$$\hat{B}_W = \sum_{i\in k}\hat{w}_i\dot{v}_i^\top, \text{ where } \hat{w}_i = \underset{w\in\mathbb{R}^k}{\arg\min}\|X_{\dot{v}_i}\hat{B}_X^\top w - Y_{\dot{v}_i}\|_2,$$

then we have with probability at least $1-\delta$,

$$\|\hat{B}_W - \dot{B}_W\| = \|\hat{B}_W - B_W\dot{V}\dot{V}^\top\| \leq 2\sqrt{k}\sin(\hat{B}_X, B_X)\|\dot{B}_W\| + \sqrt{\frac{1}{n}}(k + \log(2/\delta)^{\frac{1}{4}}|\dot{V}|^{\frac{1}{4}}$$

Proof. From [8], we get that the explicit form of $\hat{w}_i$, which is the estimation of actual $B_w\dot{v}_i$ as

$$\left(\hat{B}_X X_{\dot{v}_i}^\top X_{\dot{v}_i}\hat{B}_X^\top\right)^{-1}\hat{B}_X X_{\dot{v}_i}^\top X_{\dot{v}_i}B_X^\top B_X\dot{v}_i + \left(\hat{B}_X X_{\dot{v}_i}^\top X_{\dot{v}_i}\hat{B}_X^\top\right)^{-1}\hat{B}_X X_{\dot{v}_i}^\top\xi_w$$

By abusing notation a little bit, here we use subscription i to denote the items that associate the task encoded by $\dot{v}_i$. Therefore, we have

$$\begin{aligned}
\hat{B}_W &= \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top X_i B_X^\top B_W \dot{v}_i \dot{v}_i^\top + \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top \\
&= \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top X_i \left(\hat{B}_X^\top \hat{B}_X + \hat{B}_{X,\perp}^\top \hat{B}_{X,\perp}\right) B_X^\top B_W \dot{v}_i \dot{v}_i^\top \\
&\quad + \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top \\
&= \hat{B}_X B_X^\top \dot{B}_W + \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top X_i \hat{B}_{X,\perp}^\top \hat{B}_{X,\perp} B_X^\top B_W \dot{v}_i \dot{v}_i^\top \\
&\quad + \sum_{i=1}^{d} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top
\end{aligned}$$

And the estimation difference between $B_W, \hat{B}_W$ can be decomposed into three parts

$$\begin{aligned}
\|\dot{B}_W - \hat{B}_W\| &\le \| \left(\hat{B}_X B_X^\top - I_k\right) \dot{B}_W \| \\
&\quad + \| \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top X_i \hat{B}_{X,\perp}^\top \hat{B}_{X,\perp} B_X^\top B_W \dot{v}_i \dot{v}_i^\top \| \\
&\quad + \| \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top \| \\
&\le \| \left(\hat{B}_X B_X^\top - I_k\right) \| \|\dot{B}_W\| \\
&\quad + \max_i \| \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top X_i \hat{B}_{X,\perp}^\top \hat{B}_{X,\perp} B_X^\top \| \| \sum_{i=1}^{t} B_W \dot{v}_i \dot{v}_i^\top \| \\
&\quad + \| \sum_{i=1}^{t} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top \|
\end{aligned}$$

By using Lemma B.3 and Lemma B.4, we can bound the first two terms by

$$2\sqrt{k} \sin(\hat{B}_X, B_X) \|\dot{B}_W\|$$

Now we are going to bound the last term which is the noise term.

$$\begin{aligned}
&\| \sum_{i=1}^{|\dot{V}|} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top \|^2 \\
&= \lambda_{\max} \left(\sum_{i=1}^{|\dot{V}|} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top \right) \left(\sum_{i=1}^{|\dot{V}|} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \dot{v}_i^\top \right)^\top \\
&\le \lambda_{\max} \left(\sum_{i=1}^{|\dot{V}|} \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i \xi_i^\top X_i \hat{B}_X^\top \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \right)
\end{aligned}$$

Note that, $x_i \sim \mathcal{N}(0, I_d)$ and

$$\begin{aligned}
\left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top \xi_i &\sim \mathcal{N}\left(0, \left(\left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \hat{B}_X X_i^\top X_i \hat{B}_X^\top \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1} \right)\right) \\
&\sim \mathcal{N}\left(0, \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1}\right)
\end{aligned}$$

Therefore, by the concentration inequality of the covariance matrix, we have, w.h.p $1-\delta$,

$$\lambda_{\max}\left(\sum_{i=1}^{|\dot{V}|}\left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1}\hat{B}_X X_i^\top \xi_i \xi_i^\top X_i \hat{B}_X^\top \left(\hat{B}_X X_i^\top X_i \hat{B}_X^\top\right)^{-1}\right) \leq \frac{1}{n}\sqrt{(k+\log(2/\delta)|\dot{V}|}$$

Combining everything above, we have the final bound. □

Lemma B.3. *Given $\hat{B}_X, B_X$ are orthonormal matrices, as well as $\mathbb{E}[xx^T] = I_{d_X}$ for all tasks w, we have*

$$\|I_k - \hat{B}_X B_X^\top\| \leq \mathcal{O}\left(\sqrt{k}\sin(\hat{B}_X, \hat{B}_X)\right)$$

Proof. Denote $B_X \hat{B}_X^\top = UDV^\top$, by definition, we have $D = \text{diag}(\cos\theta_1, \cos\theta_2, \ldots, \cos\theta_k)$ from the largest singular value to minimum singular value and $\sin\theta_k \leq \sin(\hat{B}_X, \hat{B}_X)$. Therefore we have,

$$\text{Tr}(\hat{B}_X B_X^\top) \geq k\sqrt{1-\sin^2(\hat{B}_X, B_X)} \geq k - k\sin^2(\hat{B}_X, B_X)$$

And

$$\begin{aligned}
\|I_k - \hat{B}_X B_X^\top\|^2 &= \lambda_{\max}\left(I_k - \hat{B}_X B_X^\top\right)^\top\left(I_k - \hat{B}_X B_X^\top\right) \\
&\leq \text{Tr}\left(I_k - \hat{B}_X B_X^\top\right)^\top\left(I_k - \hat{B}_X B_X^\top\right) \\
&\leq \text{Tr}\left(I_k + \left(\hat{B}_X B_X^\top\right)^\top \hat{B}_X B_X^\top - \left(\hat{B}_X B_X^\top\right)^\top - \hat{B}_X B_X^\top\right) \\
&\leq 2k - 2k + 2k\sin^2(\hat{B}_X, B_X) \leq 2k\sin^2(\hat{B}_X, B_X)
\end{aligned}$$

□

Lemma B.4 (Restate from [11])**.** *Given $\hat{B}_X, B_X$ are orthonormal matrices, as well as $\mathbb{E}[xx^T] = I_{d_X}$ for any fixed task w, we have*

$$\|\left(\hat{B}_X X_w^\top X_w \hat{B}_X^\top\right)^{-1}\hat{B}_X X_w^\top X_w \hat{B}_{X,\perp}^\top \hat{B}_{X,\perp} B_X^\top\| \leq \sin(\hat{B}_X, \hat{B}_X)$$

Proof. Here we follow the same proof step as in [11]. (Bound on the second error term in Lemma 19)

$$\begin{aligned}
&\|\left(\hat{B}_X X_w^\top X_w \hat{B}_X^\top\right)^{-1}\hat{B}_X X_w^\top X_w \hat{B}_{X,\perp}^\top \hat{B}_{X,\perp} B_X^\top\| \\
&\leq \|\left(\hat{B}_X X_w^\top X_w \hat{B}_X^\top\right)^{-1}\hat{B}_X X_w^\top X_w \hat{B}_{X,\perp}^\top\|\sin(\hat{B}_X, \hat{B}_X) \\
&\leq \sin(\hat{B}_X, \hat{B}_X)
\end{aligned}$$

□

B.3 Lemmas about the properties of W'

Lemma B.5.

$$\textit{rank}(\hat{B}_W W W' \hat{B}_W^\top) \leq \textit{rank}(B_W \mathbb{E}_{\nu_{target}}[w_0 w_0^\top] B_W^\top)$$

Proof. By using Welys inequality, we have for any eigenvalue $i \in [k]$,

$$\begin{aligned}
&|\lambda_i\left(\hat{B}_W^{\text{target}}\mathbb{E}[w_0 w_0^\top](\hat{B}_W^{\text{target}})^\top\right) - \lambda_i\left(B_W^{\text{target}}\mathbb{E}[w_0 w_0^\top](B_W^{\text{target}})^\top\right)| \\
&\leq \|\hat{B}_W^{\text{target}}\mathbb{E}[w_0 w_0^\top](\hat{B}_W^{\text{target}})^\top - B_W^{\text{target}}\mathbb{E}[w_0 w_0^\top](B_W^{\text{target}})^\top\| \\
&\leq \|\hat{B}_W^{\text{target}}(\hat{B}_W^{\text{target}})^\top - B_W^{\text{target}}(B_W^{\text{target}})^\top\| \\
&\leq 2\|\hat{B}_W^{\text{target}} - B_W^{\text{target}}\|\|B_W^{\text{target}}\| \\
&\leq \left(2k^{\frac{3}{2}}\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{source}}\|^2 + 2\sqrt{\frac{k}{\dot{n}_{\text{target}}}}\right)\|B_W^{\text{target}}\|
\end{aligned}$$

where the last inequality comes from Lemma B.2 and the fact $\sin(\hat{B}_X, B_X) \leq \widetilde{\mathcal{O}}\left(k\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{source}}\|\right)$. Therefore, for all the $i \geq k^*$,

$$\lambda_i\left(\hat{B}_W^{\text{target}}\mathbb{E}[w_0 w_0^\top](\hat{B}_W^{\text{target}})^\top\right) \geq \left(2k\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{source}}\|^2 + 2\sqrt{\frac{k}{\dot{n}_{\text{target}}}}\right)\|B_W^{\text{target}}\|$$

Clipping those non-significant directions leads to the result. □

Lemma B.6. *Define* $W_i^* = \arg\min_v \|v\|_2 \quad$ *, s.t.*$\hat{B}_W^{source}v = \hat{u}_i\hat{\Lambda}_i$*, we have*

$$\max_i \|W_i'\| \leq \min\{k^*, \kappa^2(B_W^{source})\}\max_i \|W_i^*\|$$

Proof. By definition of W', we have, for any W_i',

$$W_i' = \arg\min_v \|v\|_2 \quad \text{, s.t.} \hat{B}_W^{\text{source}}v = \hat{u}_i\hat{\Lambda}_i$$

where $\hat{U}, \hat{\Lambda}_i \leftarrow \text{Eig}(\mathbb{E}[\hat{B}_W^{\text{target}}w_0 w_0^\top(\hat{B}_W^{\text{target}})^\top])$. By solving this optimization, we get

$$W_i' = (\hat{B}_W^{\text{source}})^\top\left(\hat{B}_W^{\text{source}}(\hat{B}_W^{\text{source}})^\top\right)^{-1}\hat{u}_i\sqrt{\hat{\Lambda}_i}$$

and therefore,

$$\begin{aligned}
\max_i \|W_i'\|^2 &= \max_i \sqrt{\hat{\lambda}_i}\hat{u}_i^\top\left(\hat{B}_W^{\text{source}}(\hat{B}_W^{\text{source}})^\top\right)^{-1}(\hat{B}_W^{\text{source}})(\hat{B}_W^{\text{source}})^\top\left(\hat{B}_W^{\text{source}}(\hat{B}_W^{\text{source}})^\top\right)^{-1}\hat{u}_i\hat{\Lambda}_i \\
&= \max_i \sqrt{\hat{\lambda}_i}\hat{u}_i^\top\left(\hat{B}_W^{\text{source}}(\hat{B}_W^{\text{source}})^\top\right)^{-1}\hat{u}_i\sqrt{\hat{\Lambda}_i} \\
&\lessapprox \max_i \hat{\lambda}_i\hat{u}_i^\top\left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1}\hat{u}_i
\end{aligned}$$

where the last inequality comes from Lemma B.1. Similarly, the ground truth W^* can be represented as

$$\begin{aligned}
\max_i \|W_i^*\|^2 &= \max_i \lambda_i u_i^\top\left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1}u_i \\
&\text{where, } \mathbb{E}_{w_0}\left[B_W^{\text{target}}w_0 w_0^\top(B_W^{\text{target}})^\top\right].
\end{aligned}$$

and denote $H = \hat{U}\hat{\Lambda}\hat{U}^\top - \mathbb{E}_{w_0}\left[B_W^{\text{target}}w_0 w_0^\top(B_W^{\text{target}})^\top\right]$.

Now we are now going to upper bound $\max_i \|W_i'\|$ in terms of $\max_i \|W_i^*\|$. Suppose $j = \arg\max \|W_i'\|$ and $B_W^{\text{target}} = U\Lambda U^\top$.

Firstly, we will lower bound the $\hat{\lambda}_i$. Given $\|\mathbb{E}_{w_0}\left[B_W^{\text{target}}w_0 w_0^\top(B_W^{\text{target}})^\top\right]\| \leq \frac{1}{2k}$, we can always found an $\|W_i'\|^2 \geq \frac{1}{2k\sigma_{\max}^2(B_W^{\text{source}})}$. Therefore, we have

$$\hat{\lambda}_j \geq \frac{1}{2k\kappa(B_W^{\text{source}})}$$

Then we consider the following two cases.

(Case 1) When $\kappa(B_W^{\text{source}})$ is small: By Wely's inequality, there always exists some u_m, λ_m that $\hat{\lambda}_j \leq \mathcal{O}(\lambda_m)$. Therefore,

$$\begin{aligned}
\hat{\lambda}_j\hat{u}_j^\top\left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1}\hat{u}_j &\leq \hat{\lambda}_j u_m^\top\left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1}u_m\kappa(B_W^{\text{source}})^2 \\
&\leq \lambda_m u_m^\top\left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1}u_m\kappa(B_W^{\text{source}})^2 \\
&\leq \max_i \|W_i^*\|^2\kappa(B_W^{\text{source}})^2
\end{aligned}$$

(Case 2) When $\kappa(B_W^{\text{source}})$ is large: Decompose $\hat{B}_W^{\text{target}}W'(W')^\top(\hat{B}_W^{\text{target}})^\top$ as follows

$$\begin{aligned}
\hat{B}_W^{\text{target}}W'(W')^\top(\hat{B}_W^{\text{target}})^\top &= \hat{U}_0\hat{\Lambda}_0\hat{U}_0^\top + \hat{U}_1\hat{\Lambda}_1\hat{U}_1^\top \\
&\text{where, } \hat{u}_j \in \hat{U}_0 \text{ and } \lambda_{\min}(\hat{\Lambda}_0) - \lambda_{\min}(\hat{\Lambda}_0) \geq \frac{1}{4}\hat{\lambda}_j
\end{aligned}$$

Correspondingly, we can decompose $\mathbb{E}_{w_0} B_W^{\text{target}} w_0 w_0^\top (B_W^{\text{target}})^\top$ as the same shape

$$\mathbb{E}_{w_0}\left[B_W^{\text{target}} w_0 w_0^\top (B_W^{\text{target}})^\top\right] = U_0\Lambda_0 U_0^\top + U_1\Lambda_1 U_1^\top$$

By using Davis-Kahn theorem, we have

$$\|U_1^\top \hat{u}_j\| \le \|U_1^\top \hat{U}_0\| \le \frac{\|U_1^\top H \hat{U}_0\|}{\frac{1}{4}\hat{\lambda}_j} \lessapprox k\|H\|\kappa(B_W^{\text{source}})$$

Since

$$\begin{aligned}
\|H\| &\le \bar{\gamma} + \left\|\mathbb{E}_{w_0}\left[B_W^{\text{target}} w_0 w_0^\top (B_W^{\text{target}})^\top\right] - \mathbb{E}_{w_0}\left[\hat{B}_W^{\text{target}} w_0 w_0^\top (\hat{B}_W^{\text{target}})^\top\right]\right\| \\
&\le 2\left\|\mathbb{E}_{w_0}\left[B_W^{\text{target}} w_0 w_0^\top (B_W^{\text{target}})^\top\right] - \mathbb{E}_{w_0}\left[\hat{B}_W^{\text{target}} w_0 w_0^\top (\hat{B}_W^{\text{target}})^\top\right]\right\| \\
&\le 2\left(2k\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{source}}\|^2 + 2\sqrt{\frac{k}{\dot{n}_{\text{target}}}}\right)\|B_W^{\text{target}}\|
\end{aligned}$$

then we have

$$\|U_1^\top \hat{u}_j\| \lessapprox 8k\left(2k\sqrt{\frac{d_X}{n_1}}\|B_W^{\text{source}}\|^2 + 2\sqrt{\frac{k}{\dot{n}_{\text{target}}}}\right)\|B_W^{\text{target}}\|\kappa(B_W^{\text{source}}) \le \frac{1}{2}$$

which suggests $\|U_0^\top \hat{u}_j\| = \|[U_0, U_1]^\top \hat{u}_j - [0, U_1]^\top \hat{u}_j\| \ge 1 - \|U_1^\top \hat{u}_j\| \ge \frac{1}{2}$. Therefore, there exists some u_m as one of the columns of U_0 that such $u_m^\top \hat{u}_j \le \mathcal{O}(\sqrt{\frac{1}{k^*}})$. And therefore, we have

$$\begin{aligned}
\hat{\lambda}_j \hat{u}_j^\top \left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1} \hat{u}_j &\le k^*\lambda_m(\hat{u}_m^\top \hat{u}_j)\hat{u}_j^\top \left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1} \hat{u}_j(\hat{u}_j^\top \hat{u}_m) \\
&\le k^*\lambda_m \hat{u}_m^\top \left(B_W^{\text{source}}(B_W^{\text{source}})^\top\right)^{-1} \hat{u}_m \\
&\le k^* \max_i \|W_i^*\|^2
\end{aligned}$$

□

Lemma B.7.

$$\|W'W'^\top\|_* \le \mathcal{O}\left(\frac{1}{\sigma_{\min}^2(B_W^{\textit{source}})}\|B_W^{\textit{target}}\mathbb{E}_{\nu_{\textit{target}}}[w_0 w_0^\top](B_W^{\textit{target}})^\top\|_*\right)$$

Proof.

$$\begin{aligned}
\|W'W'^\top\|_* &\le \frac{1}{\sigma_{\min}^2(B_W^{\text{source}})}\|B_W^{\text{source}} W'(W')^\top (B_W^{\text{source}})^\top\|_* \\
&\le \frac{1}{\sigma_{\min}^2(B_W^{\text{source}})}\|\hat{B}_W^{\text{source}} W'(W')^\top (\hat{B}_W^{\text{source}})^\top\|_* \\
&\le \frac{1}{\sigma_{\min}^2(B_W^{\text{source}})}\|\hat{B}_W^{\text{target}}\mathbb{E}_{\nu_{\text{target}}}[w_0 w_0^\top](\hat{B}_W^{\text{target}})^\top\|_* \\
&\le \frac{1}{\sigma_{\min}^2(B_W^{\text{source}})}\|B_W^{\text{target}}\mathbb{E}_{\nu_{\text{target}}}[w_0 w_0^\top](B_W^{\text{target}})^\top\|_*
\end{aligned}$$

□

B.4 Sample complexity analysis – Formal version of Theorem 4.1

Theorem B.3 (Formal theorem)**.** *By running Algo. 2, in order to let $ER(\hat{\phi}_X, \nu_{\textit{target}}) \le \varepsilon^2$ with probability $1-\delta$, where $\delta \ge (d_X k)^{10}$, then the number of source samples $n_{\textit{source}}$ is at most*

$$\begin{aligned}
&\widetilde{\mathcal{O}}\left(\sigma^2(k^*)^2 \min\{\kappa(B_W^{\textit{source}}, k^*\} \max_i \|W_j^*(i)\|_2^2 k d_X \varepsilon^{-2}\right) \\
&+ \widetilde{\mathcal{O}}\left(\varepsilon^{-\frac{4}{3}} k^{\frac{7}{3}} d_W^{\frac{2}{3}} d_X \left(k^{\frac{2}{3}} d_W^{\frac{1}{3}} \underline{\sigma}^{-\frac{4}{3}} + \bar{\kappa}^2 \underline{\sigma}^{-\frac{2}{3}}\right)\right) \\
&+ \widetilde{\mathcal{O}}\left(\bar{\kappa}^2\sqrt{k}\left(k^2 d_X \bar{\kappa}^2 + \frac{d_W^{\frac{3}{2}}}{\underline{\sigma}^2}\sqrt{k + \log(d_W/\delta)}\right)\right)
\end{aligned}$$

Here $k^* = rank(\mathbb{E}_{w_0 \sim \nu_{target}} B_W w_0 w_0^\top B_W^\top)$ *represents the effective dimension of target and*

$$W_i^* = \underset{w \in \mathcal{W}_{source}}{\arg\min} \|w\|_2 \quad s.t \quad B_W^{source} w = u_i \sqrt{\lambda_i} \text{ where } U, \Lambda \leftarrow Eig(\mathbb{E}_{w_0 \sim \nu_{target}} B_W w_0 w_0^\top B_W^\top).$$

as long as,

$$\dot{n}_{target} \geq \widetilde{\Omega}\left(\varepsilon^{-\frac{4}{3}} (k^*)^{\frac{2}{3}} \left(d_W^{\frac{1}{2}} \underline{\sigma}^{-\frac{4}{3}} + k^{-\frac{2}{3}} d_W^{\frac{1}{6}} \bar{\kappa}^2 \underline{\sigma}^{-\frac{1}{3}}\right) \sqrt{k + \log(d_W/\delta)}\right)$$
$$n_{target} \geq \widetilde{\Omega}\left((k + \log(1/\delta))\varepsilon^{-2}\right)$$

Proof. By setting the target excess risk ε^2 and the generalization guarantees in Theorem B.2, we have

$$\sigma^2 \left(k d_X \log((\kappa N_i)/d_W) + \log \frac{1}{\delta}\right) k^* \epsilon_j^2 = \varepsilon^2 \tag{3}$$

After some rearrangement, we can directly have the guarantees for $n_1^j, n_0, \dot{n}_{\text{target}}, n_{\text{target}}$. Sum over the epoch gives our desired result. Now we will focus on n_2^j.

$$\begin{aligned} n_2^j &\leq \widetilde{\mathcal{O}}(k^* \max_i \|W_j'(i)\|_2^2 \epsilon_j^{-2}) \\ &\leq \widetilde{\mathcal{O}}(k^*(\kappa(B_W^{\text{source}} + k^*)) \max_i \|W_j^*(i)\|_2^2 \epsilon_j^{-2}) \\ &\leq \widetilde{\mathcal{O}}\left(\sigma^2 (k^*)^2 \min\{\kappa(B_W^{\text{source}}, k^*\} \max_i \|W_j^*(i)\|_2^2 \left(k d_X + \log(1/\delta)\right)\right) \varepsilon^{-2}\right) \end{aligned}$$

where the first inequality comes from the definition and the second inequality comes from the Lemma B.6.

Finally, by union bounding on the $1 - \delta$ from Theorem B.2 and the event $\mathcal{E}_{\text{offline 1}}$ over all the epochs, we get the target result. □

B.5 Algorithms in [23] is a special case of Algo. 2

Specifically, in this paper, we aim to minimize.

$$(\hat{B}_W^{\text{target}} w_0)^\top (\hat{B}_W^{\text{source}} Q (\hat{B}_W^{\text{source}})^\top)^{-1} (\hat{B}_W^{\text{target}} w_0), \text{ where } \sum_i q_i = 1 \text{ [eq. 1]}$$

which can be equivalently written as

$$\text{trace}(V^\top Q V)^{-1} (V^\top u u^\top V) = \sum_i q_i^{-1} v_i^2$$

where V comes from the eigendecomposition of $\hat{B}_W^{\text{source}} = U D V^\top$ and u can be anything satisfying $\hat{B}_W^{\text{source}} v = \hat{B}_W^{\text{target}} w_0$.

Therefore, if we assume $q_i = |u_i|^\alpha / \sum_i |u_i|^\alpha$ for $\alpha > 0$, then [8] is equivalent to choosing $\alpha = 2$ and [23] is equivalent to choosing $\alpha = 1$. It is easy to see that $\alpha = 1$ is the optimal solution.

B.6 More interpretation on results

Lemma B.8. *When = the target task is uniformly spread* $\|\mathbb{E}_{w_0 \sim \nu_{target}} B_W w_0 w_0^\top B_W^\top\| = \frac{1}{k}$ *and the task representation is well-conditioned* $\sigma_{\min}^2(B_W^{source}) = \frac{d_W}{k}$*, we have*

$$\|W_i^*\|_2^2 = \frac{1}{d_W}$$

Proof. Do a svd decomposition on the B_W^{source} gives $\sqrt{\frac{d_W}{k}} U_1 V_1^\top$. For any i, let w satisfies

$$\sqrt{\frac{d_W}{k}} U_1 V_1^\top w = \sqrt{\frac{1}{k}} u_i$$

Rearranging the above equality gives $V_1^\top w = \sqrt{\frac{1}{d_W}} U_1^\top u_i$. Because $W_i = \arg\min \|w\|_2$ satisfy the above constraints, we have

$$\|W_i^*\|_2^2 = \|V_1^\top w\|_2^2 = \left\| \sqrt{\frac{1}{d_W}} U_1^\top u_i \right\|_2^2 = \frac{1}{d_W}$$

□

Lemma B.9. *Let*

$$\nu_1 = \arg\min_{\nu} \|\nu\|_1 \; s.t \; B_W \nu = B_W w_0$$
$$\nu_2 = \arg\min_{\nu} \|\nu\|_2 \; s.t \; B_W \nu = B_W w_0$$

Then $\|\nu_1\|_1^2 \geq \|\nu_2\|_2^2$.

Proof.

$$\|\nu_1\|_1^2 \geq \|\nu_1\|_2^2 \geq \|\nu_2\|^2$$

□

C Results and analysis for target-agnostic

C.1 Algorithm for target-agnostic

Algorithm 3 Target-agnostic algorithm for benign source space

1: **Inputs:** Target probability δ, $\bar{\kappa}, \underline{\sigma}$. Some constant $\beta_1, \beta_2, \beta_3$. Others same as Algo. 1.
2: Set q_0 as $q_0(e_t) = \frac{1}{d_W}, \forall t \in d_W$, and$q_0(w) = 0$ otherwise
3: Set $n_0 = \beta_1 \beta_1 \bar{\kappa}^2 \left(k^3 d_X \bar{\kappa}^2 + d_W^{\frac{3}{2}} \underline{\sigma}^{-2} \sqrt{k + \log(1/\delta)} \right)$. Collect $n_0 q_0(w)$ data for each task denoted as $\{Z_w\}_{w|q(w)\neq 0}$
4: Update $\hat{B}_X \leftarrow \mathcal{O}^X_{\text{offline }1}(\{Z_w\}_{w|q_0(w)\neq 0})$ and $\hat{B}_W^{\text{source}} \leftarrow \mathcal{O}^W_{\text{offline}}(\{Z_w\}_{w|q_0(w)\neq 0}, \hat{B}_X)$
5: Compute q_1 as $q_1(v_i) = \frac{1}{k}, \forall i \in k$, and $q_0(w) = 0$ otherwise. Here v_i is the i-th vector of V, where $U, D, V \leftarrow \text{SVD}(\hat{B}_W^{\text{source}})$
6: For any given budget n_1, collect $n_1 q_1(w)$ data for each task denoted as $\{Z_w\}_{w|q_1(w)\neq 0}$.
7: Update $\hat{B}_X \leftarrow \mathcal{O}^X_{\text{offline }2}(\{Z_w\}_{w|q_1(w)\neq 0})$, $\hat{B}_W^{\text{source}} \leftarrow \mathcal{O}^W_{\text{offline}}(\{Z_w\}_{w|q_1(w)\neq 0}, \hat{B}_X)$
8: **Return** $\tilde{B}_X$

C.2 Results and analysis

Theorem C.1. *In order to get* $\text{ER}(\hat{B}_X, \nu_{target}) \leq \varepsilon^2$*, we have w.h.p* $1-\delta$*, source samples complexity is at most*

$$\widetilde{\mathcal{O}}\left(\frac{k^2 d_X Tr(B_W \mathbb{E}[w_0 w_0^\top] B_W^\top)}{\sigma_k^2(B_W^{source})} \varepsilon^{-2} \right) + \widetilde{\mathcal{O}}\left(\bar{\kappa}^2 \left(k^2 d_X \bar{\kappa}^2 + \frac{d_W^{\frac{3}{2}}}{\underline{\sigma}^2} \sqrt{k + \log(d_W/\delta)} \right) \right)$$

as long as,

$$\dot{n}_{target} \geq \widetilde{\Omega}\left(\varepsilon^{-\frac{4}{3}} (k^*)^{\frac{2}{3}} \left(d_W^{\frac{1}{2}} \underline{\sigma}^{-\frac{4}{3}} + k^{-\frac{2}{3}} d_W^{\frac{1}{6}} \bar{\kappa}^2 \underline{\sigma}^{-\frac{1}{3}} \right) \sqrt{k + \log(d_W/\delta)} \right)$$
$$n_{target} \geq \widetilde{\Omega}\left((k + \log(1/\delta)) \varepsilon^{-2} \right)$$

Proof. Again from Section B.2.1, we have w.h.p at least $1-\delta$

$$\text{ER}(\hat{B}_X, \nu_{\text{target}}) \lessapprox \sigma^2 n_{\text{target}} \left(k d_X + \log(1/\delta) \right) \text{Tr}\left(\left((B_W^{\text{source}}) \left(\sum_{w \in \mathcal{S}} n_w w w^\top \right) (B_W^{\text{source}})^\top \right)^{-1} B_W \left(\mathbb{E}_{\nu_{\text{target}}} w w^T \right) B_W^\top \right)$$
$$+ \frac{k + \log(1/\delta)}{n_{\text{target}}}$$

then by using similar steps in Section B.2.2, we have

$$\begin{aligned}
&\mathrm{Tr}\left(\left((B_W^{\text{source}})\left(\sum_{w\in\mathcal{S}} n_w w w^\top\right)(B_W^{\text{source}})^\top\right)^{-1} B_W\left(\mathbb{E}_{\nu_{\text{target}}} w w^T\right) B_W^\top\right) \\
&\le \|\square\|\mathrm{Tr}(B_W \mathbb{E}[w_0 w_0^\top] B_W^\top) \\
&\le \frac{k}{n_1 \sigma_k^2(B_W^{\text{source}})}\mathrm{Tr}(B_W \mathbb{E}[w_0 w_0^\top] B_W^\top)
\end{aligned}$$

and therefore,

$$\mathrm{ER}(\hat{B}_X, \nu_{\text{target}}) \le \widetilde{\mathcal{O}}\left(\frac{k^2 d_X}{n_1 \sigma_k^2(B_W^{\text{source}})}\mathrm{Tr}(B_W \mathbb{E}[w_0 w_0^\top] B_W^\top)\right) + \widetilde{\mathcal{O}}\left(\frac{k+\log(1/\delta)}{n_{\text{target}}}\right)$$

Rearranging the inequality gives the final bound. □

C.3 Compare to previous passive learning and the target-aware one

Again we want to compare this result with the previous one.

Comparison with passive learning. We first consider the cases in their paper that the target task is uniformly spread $\|\mathbb{E}_{w_0\sim\nu_{\text{target}}} B_W w_0 w_0^\top B_W^\top\| = \frac{1}{k}$. (See detailed setting in Section 4)

- When the task representation is well-conditioned $\sigma_{\min}^2(B_W^{\text{source}}) = \frac{d_W}{k}$. We have a passive one as $\widetilde{\mathcal{O}}(k d_X \varepsilon^{-2})$ while the target-agnostic active one $\widetilde{\mathcal{O}}(k d_X \frac{k^2}{d_W}\varepsilon^{-2})$.
- Otherwise, we consider the extreme case that $\sigma_{\min}^2(B_W^{\text{source}}) = 1$. We have passive one $\widetilde{\mathcal{O}}(d_X d_W \varepsilon^{-2})$ while the target-agnostic active one $\widetilde{\mathcal{O}}(k^2 d_X \varepsilon^{-2})$. Note this is better than the $\widetilde{\mathcal{O}}(k^3 d_X \varepsilon^{-2})$ in the target-aware case.

These two results indicate that when the targets are uniformly spread, target-agnostic AL can perform even better than the target-aware. But we want to emphasize that whether it is uniformly spread or not is unknown to the learner. Even $\kappa\left(\mathbb{E}_{w_0\sim\nu_{\text{target}}}[w_0 w_0^\top]\right) = 1$ can leads to ill-conditioned $B_W \mathbb{E}_{w_0\sim\nu_{\text{target}}}[w_0 w_0^\top] B_W^\top$.

We then consider the single target w_0 case.

- With well-conditioned B_W, the passive one now has sample complexity $\mathcal{O}(k^2 d_X \varepsilon^{-2})$ while the active gives a strictly improvement $\mathcal{O}(\frac{k^3 d_X}{d_W}\varepsilon^{-2})$.
- With ill-conditioned B_W where $\sigma_{\min}(B_W) = 1$ and $\max_i \|W_i^*\| = 1$, that is, only a particular direction in source space contributes to the target. The Passive one now has sample complexity $\mathcal{O}(k d_X d_W \varepsilon^{-2})$ while our target-agnostic active one has $k^2 d_X \varepsilon^{-2}$.

These two results indicate that the target-agnostic approach gives a worse bound when the targets are not well-spread, which meets our intuition since the target-agnostic tends to learn uniformly well over all the levels. But it can still perform better than the passive one under the discrete case, which again indicates the necessity of considering the continuous setting.

Save task number. Again when ignoring the short-term initial warm-up stage, we only require maintaining $\widetilde{\mathcal{O}}(k)$ number of source tasks.

D Limitations from the theoretical perspective

Here we list some open problems from the theoretic perspective. We first list some room for improvements under the current setting

- **Not adaptive to noise σ:** From Section B.1.1, we get $\sin(\hat{B}_X, B_X)$ scales with the noise σ, which suggests less sample number n_0, n_1 is requires to get a proper estimation of B_X. In our algorithm, however, we directly treat $\sigma = \Theta(1)$ and therefore may result in unnecessary exploration.

- Bound dependence on $\min\{\kappa^2(B_X^{\text{source}}), k\}$: This extra dependence comes from the instability (or non-uniqueness) of eigendecomposition. For example, when $\mathbb{E}_{\nu_{\text{target}}}[B_W w_0 w_0^\top B_W^\top] = \frac{1}{k} I_d$, there are infinite number of eigenvector sets. On the other hand, given a fixed B_W^{source}, current methods of obtaining W' are highly sensitive to the eigenvector sets from the target. A direct method is of course constructing a confidence bound around the estimated $\hat{B}_W^{\text{target}}$ and finding the best W' under such set. But this method is inefficient. Whether there exists some efficient method, like a regularized optimization, remains to be explored in the future.
- **Require prior knowledge of $\bar{\kappa}, \underline{\sigma}$:** Finally, can we estimate and use those parameters during the training remains to be open?

Besides that specific problem, it is always meaningful to extend this setting into more complicated geometries and non-linear/non-realizable models. Specifically,

- **More complicated geometry.** One open problem is to get guarantees when $\mathcal{W}_{\text{source}}, \mathcal{W}_{\text{target}}$ is no longer a unit ball. (e.g., eclipse). Another problem is, instead of considering the geometry of $\mathcal{W}$, we should consider the geometry of $\psi_W(\mathcal{W})$.
- **Nonlinear models.** Consider nonlinear ϕ_X, ϕ_W is always challenging. In [9, 10], they provide some guarantees under the passive by using kernel methods or considering a general model. Can we extend this to the active setting?
- **Non-realizable model.** Like many representation learning papers, we assume the existence of a shared representation, which suggests more source tasks always help. In practice, however, such representation may not exist or is more over-complicated than the candidate models we assume. Under such a misspecification setting, choosing more tasks may lead to negative transfer as shown in Figure 5 in the experiments. Can we get any theoretical guarantees under such a non-realizable setting?

E Experiment details

Here we provide detailed settings of three experiments – synthetic data, pendulum simulator, and the real-world drone dataset, as well as more experimental results as supplementary. All the experiments follow a general framework proposed in Section 3 with different implementation approaches according to different settings, which we will specify in each section below. Note that in all these experiments, we only focus on a single target.

E.1 Synthetic data

E.1.1 Settings

	bilinear	nonlinear ψ_X	nonlinear ϕ_X
target number	800, 8000	800, 8000	800, 8000
d_X	200	10	20
d_{ψ_X}	200	200	20
d_W	80	80	80
k	4	4	4
ϕ structure	random matrix	random matrix	MLP with layers [20, 20, 4]
inputs distribution	$\mathcal{N}(0, I)$	$\mathcal{N}(0, I)$	$\mathcal{N}(0, I)$
label noise variance	1	1	1

Table 3: Model used to generate the synthetic data.

Data generation We show the model and corresponding parameters used to generate the synthetic data in Table. 3. Some additional details include, 1) When generating random matrix B_X for bi-linear and unknown non-linear ψ_X, we tried different seeds (denoted as *embed_matrix_seed* in the codes) and deliberately make the matrix ill-conditioned (so $\kappa(B_W)$ is large). Because most of them behave similarly so we only present partial results here. 2) When generating random MLP for nonlinear ϕ_X, we only use the unbiased linear layer and ReLU layers.

In the main paper Table 5.2, we use target number = 8000 cases to show more contrast.

The nonlinear Fourier feature kernel ψ_X is defined as $\psi_X(x) = \cos(Ax + B)$, where $A \in \mathbb{R}^{d_{\psi_X} \times d_X}$, $B \in \mathbb{R}^{d_{\psi_X}}$ and each entry of A, B is i.i.d. Gaussian.

Training models and optimizer Here we state the details of the model used during the learning, which might be different from the model used to generate the data. Specifically, for the bi-linear and unknown non-linear ψ_X, we use the exact $\mathbb{R}^{d_{\psi_X} \times k}$ matrix structure as stated in the theorem. For the nonlinear ϕ_X, we use a slightly larger MLP with layers [20, 20, 20, 4] compared to the model used to generate the data to further test the adaptivity of our algorithm since the exact underlying structure of MLP is usually unknown in reality. As for the joint training approach, we use Adam with $lr = 0.1$ for the bi-linear and unknown non-linear ψ_X, and SGD with $lr = 0.1$ for nonlinear ϕ_X as the optimizer (The learning rate is large because this is an easy-to-learn synthetic data) We mixed all the target and source data and do joint GD-based methods on them. Notice that the goal for those experiments is not to achieve the SOTA but to have a fair comparison. So all those hyper-parameters are reasonable but not carefully fine-tuned.

Detailed implementation for AL strategy Both the input space $\mathcal{X}$ and the task space $\mathcal{W}$ of synthetic data lie perfectly in a ball and the underlying model is linear in terms of w. Therefore, we can use the almost similar algorithms as proposed in Algo 2 for target-aware and Algo. 3. We slightly adjust parameter dependence on d_X, d_W, k but the general scaling between different stages in each epoch remains the same. Another difference is that, instead of using the MLLAM as specified in Section B.1.1, we do a joint-GD since the implementation of MLLAM in a non-idealistic setting (nonlinear ϕ_X, ψ_X is unclear and challenging.)

Metrics We consider the worst-case distance between ground truth and estimate columns space $U, \hat{U}$ as $\text{dis}(U, \hat{U}) = \min_u \|u_i^\top \hat{U}\|_2$. Such distance will be used in both computing the similarity between ground truth and estimated input space $B_W, \hat{B}_W$. In addition, it will also be used in measuring the change of q_2 across each epoch so we can save task numbers by maintaining the same q_2 as long as the change is small, which we will specify in the next paragraph.

Saving task number approach. In addition to the comparison between target-agnostic AL, target-aware AL, and the passive, we also consider the saveTask case, where we reduce the number of times recomputing the q_1. Specifically, we denote $W_{j-1}, W_j \in \mathbb{R}^{d_W timesk}$ as the exploration source tasks in the previous and current epoch. And only switch to the new target-agnostic exploration set when $\text{dis}(\text{rowSpace}(B_{j-1}), \text{rowSpace}(B_j)) \leq 0.8$ where 0.8 is some heuristic threshold parameter.

E.1.2 Results

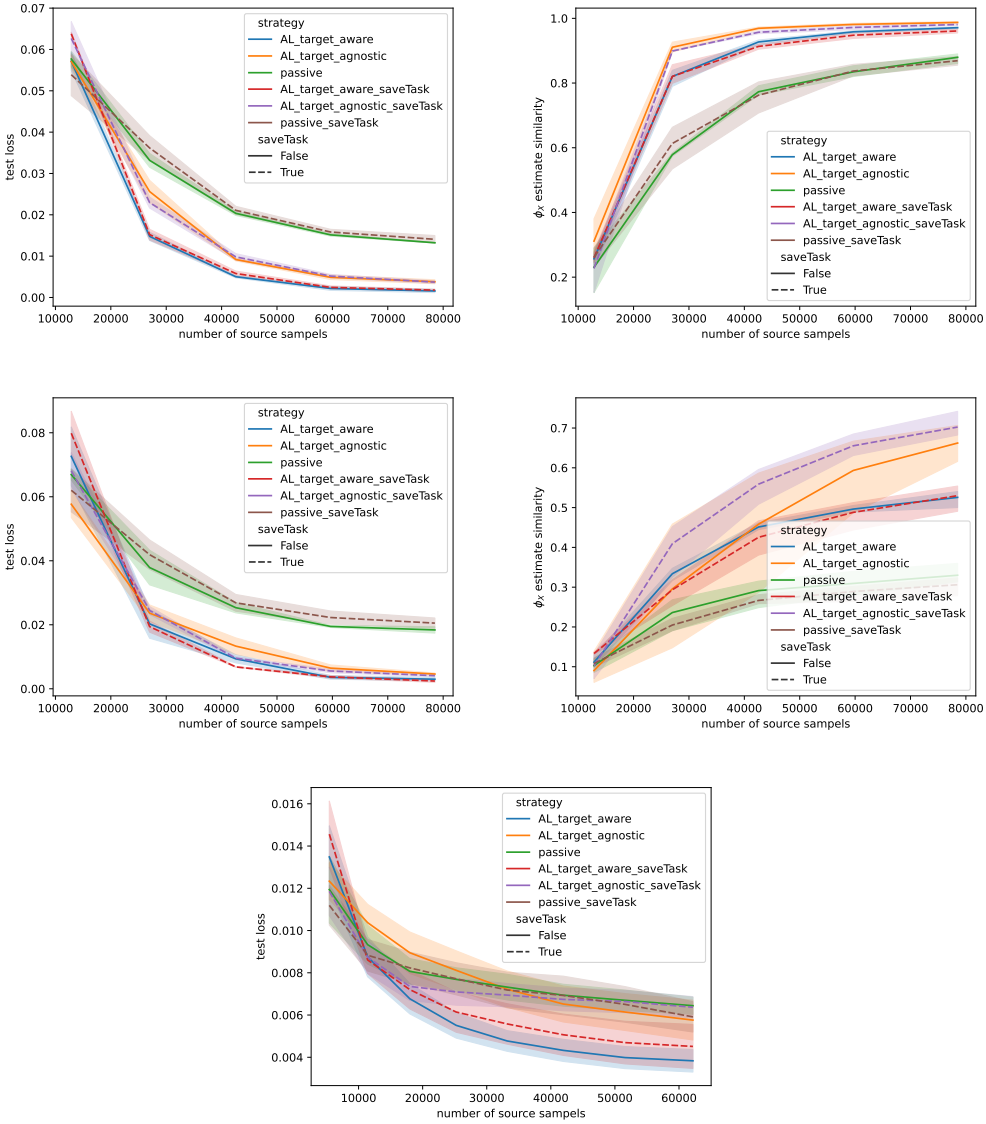

Figure 3: **Results on synthetic data with 8000 target sample** Left side presents the test loss and the right side presents the similarity between the column space of the ground truth ϕ_X and the estimated $\hat{\phi}_X$. Notice that how to measure the similarity on neural networks is unclear so we skip this result. **Top and middle:** Results of the nonlinear kernel. The target-aware AL gets the lowest test loss while the passive gets the highest. In terms of saveTask, we notice that reducing task switch number does not affect the performance a lot. From the left figure, the target-agnostic AL gets the best estimation which aligns with our design intuition that target-agnostic AL should have a universal good estimation in all directions. It is a little surprising to us that the passive one performs worst. We conjecture the reason that the GD-based oracle is not that good for joint-task training and should again have better performance when using [12, 10]. **Bottom:** Result of non-linear representation. Here we notice that the saving task strategy leads to slightly worse performance. While the target-aware AL still gives the worst test loss, the difference between passive and target-agnostic AL is small due to the complexity of the shallow net.

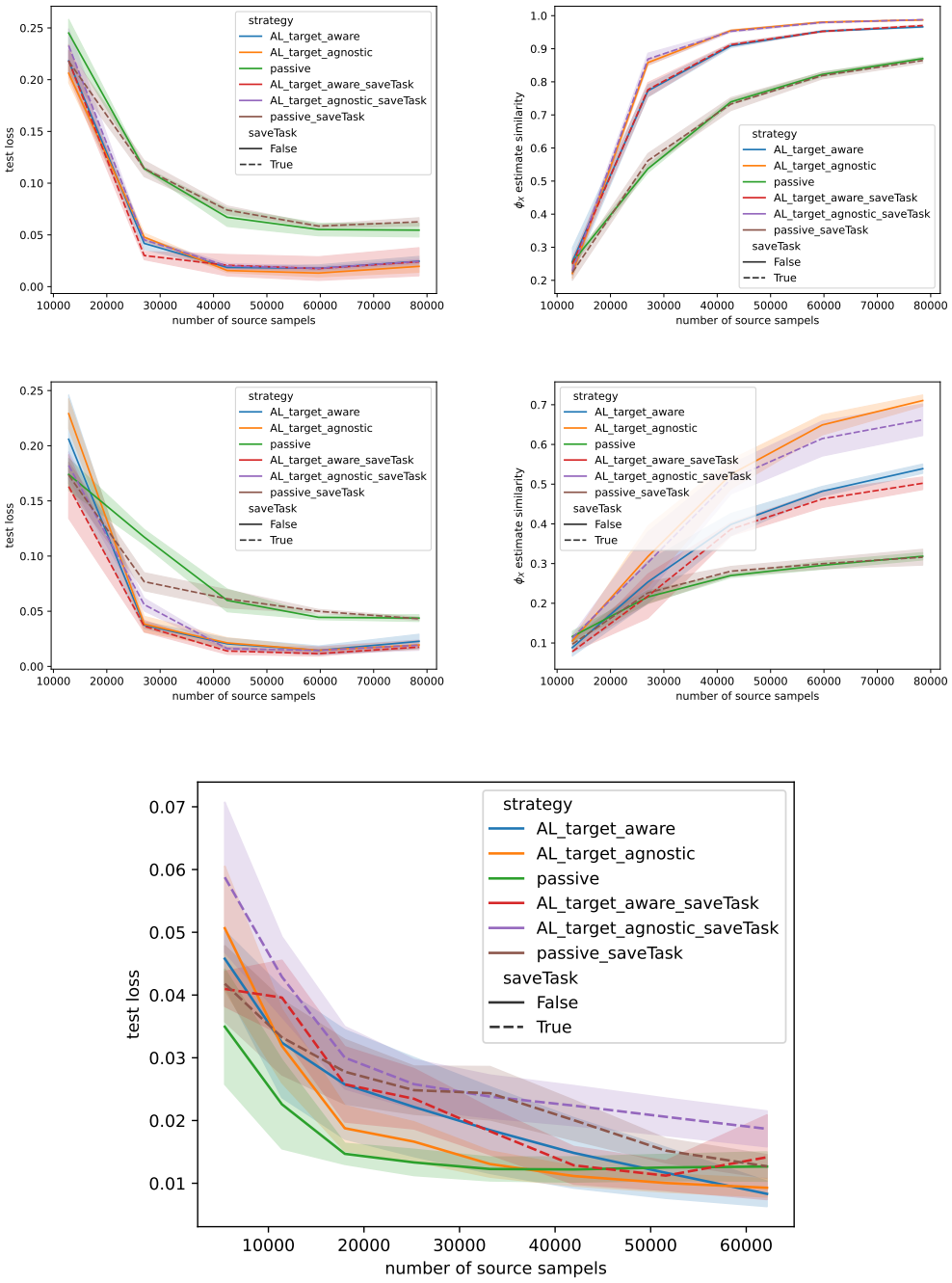

Figure 4: **Results on synthetic data with 800 target sample Top and middle:** The bilinear and nonlinear ψ_X case gives a similar performance as before. **Bottom:** For ϕ_W as a neural net, we notice here the AL does not show an advantage until the very end where the passive stops decreasing. This may suggest for nonlinear representation, more target data may be needed for a beneficial source selection compared to the bilinear ϕ.

E.2 Pendulum simulator

E.2.1 Settings

Data generation We consider the following continuous-time pendulum dynamics model adopted from [24]:

$$ml^2\ddot{\theta} - ml\hat{g}\sin\theta = u + f(\theta, \dot{\theta}, w)$$

where $\theta, \dot{\theta}, \ddot{\theta}, u$ are angle, angular velocity, angular acceleration, and control, $m, l, \hat{g}$ are mass, pole length, and the gravity estimation, and finally, f is the unknown residual dynamics term to be learned with w the environment parameter. The ground truth f is given by

$$F = \|R\|_2^2 \cdot R, R = c - \begin{bmatrix} l\dot{\theta}\cos\theta \\ -l\dot{\theta}\sin\theta \end{bmatrix}$$

$$f(\theta, \dot{\theta}, w) = \underbrace{\vec{l} \times F}_{\text{air drag}} - \underbrace{\alpha_1\dot{\theta} - \alpha_2\dot{\theta}|\dot{\theta}|}_{\text{damping}} + \underbrace{ml(g - \hat{g})\sin\theta}_{\text{gravity mismatch}}$$

$$w = [c_x, c_y, \alpha_1, \alpha_2, \hat{g}, 0 \text{ or } 1]$$

where $c = [c_x, c_y]$ is external wind, α_1, α_2 are damping coefficients and g is the true gravity.

We let $x = [\theta, \dot{\theta}]$ denote the input to f. Notice here the last element of w is a dummy feature. For the source tasks, we always have $w[6] = 0$ since all the source parameters are known. For the single target task, we have $w_{\text{actual\_target}}$ to generate the data, so $w_{\text{actual\_target}}[6] = 0$. But the learner only observes the $w_{\text{target}} = [0, 0, 0, 0, 0, 1]$, which indicates the unknown environment of the target. In the simulator, we collect data using a stochastic policy to approximate i.i.d. data distribution.

It is easy to see that f is highly nonlinear regarding x, w. Therefore we use the known nonlinear feature operator ψ to make it close to the linear model with some misspecification:

ψ_X is the Fourier feature kernel which has been defined in the synthetic data section

$\psi_W(w) = [l_x, l_y, g, \alpha_1, \alpha_2, CxCy, Cx^2, Cx^2C_y, C_x^3, Cy^2, Cy^2C_x, C_y^3, 0 \text{ or } 1]$

Other common parameters are specified in Table. 4.

target number	d_X	d_{ψ_X}	d_W	d_{ψ_W}	k	ϕ structure	inputs distribution	label noise variance
4000	2	60	13	6	8	bilinear	(See details above)	0.5

Table 4: Model parameters for pendulum simulator.

Training models and optimizer We again use the bilinear model. For the training methods, we first do joint-GD as before using AdamW with $lr = 0.01, wd = 0.05$, batch_size $= 512$. Then after joint training, we freeze the ϕ_X parts and only trained on the targets to get the non-shared embed $\phi_W(w_{\text{target}})$. Another modification is that, since we are in the misspecification setting, using data collected in stage 3 might amplify the errors when estimating the target-related source. To tackle this negative transfer learning, we only use the data collect from stage 2 in previous the epochs to compute q_3. While in the synthetic data, all data, including one from stage 3, collected in previous epochs can be used.

Detailed implementation for AL strategy The input space $\mathcal{X}$ and task space $\mathcal{W}$ of this pendulum data again lie perfectly in a ball after some normalization. Nevertheless, the underlying model is no longer linear in terms of w, which adds some extra difficulties to the optimal design on w. Here we use the adaptive sampling methods mentioned in the main paper. That is, we will iteratively sample from $\mathcal{W}_{\text{source}}$ and find the ones that minimize follows.

$$\min_{\{w_i\} \in \mathcal{W}_{\text{source}}} \|\hat{B}_{W,j}^{\text{source}} \psi_W(w_i) - u_i\sqrt{\lambda_i}\|$$

where $u_i\sqrt{\lambda_i}$ is defined in line 9. Other parts of the algorithm can still be implemented as in the synthetic data section.

Using learned f for control To show that a better dynamics model can transfer to better control performance, we deploy the following nonlinear controller $\pi(x, \hat{f})$ as a function of $\hat{f}$ (prediction result of f in the target task):

$$u = -ml\hat{g}\sin\theta - \hat{f}(\theta, \dot{\theta}) - ml^2(K_P\theta + K_D\dot{\theta})$$

Here we focus on the regulation task, i.e., $\|x\| \to 0$. It is worth noting that the above controller is guaranteed to be exponentially stable: $\|x\| \to \eta$ exponentially fast, where η is an error ball whose size is proportional to $\|f - \hat{f}\|_\infty$.

E.2.2 Results

In the main paper, we use the unobservable actual target as $[0, 0, 1, 0.5, 0, 0]$. Here we give more results in Figure. 5

E.3 Real-world drone flight dataset

E.3.1 Settings

The training model and optimizer Here we use two layer MLP model as specified below. For the training methods, we do joint-GD as before using AdamW with $lr = 0.005$ and batch_size$= 1000$. Other common parameters are specified in Table. 5.

target number	d_X	d_{ψ_X}	d_W	d_{ψ_W}	k	ϕ structure
500	11	11	18 one-hot	18	2	MLP with hidden layers $[11, 2]$

Table 5: Model parameters for drone dataset.

Data generation We use the same data as stated in the main paper.

Detailed implementation for AL strategy Unlike the previous two settings where the task space $\mathcal{W}$ is continuous, here we consider a discrete task space. Therefore the Algo. 2 no longer works. Therefore, here we use a similar technique as the Algorithm proposed in [8], which can be seen as a special case under the general Algo. 1. We want to emphasize that this choice is due to the limitation of real-world datasets, i.e., we can not arbitrarily query w to sample, and the main purpose is to show the potential of such a framework in real-world robotics applications.

E.3.2 Results

In the main paper, we provide the result when assuming a bilinear underlying model. Here we further show the effectiveness of our methods under nonlinear ϕ_X.

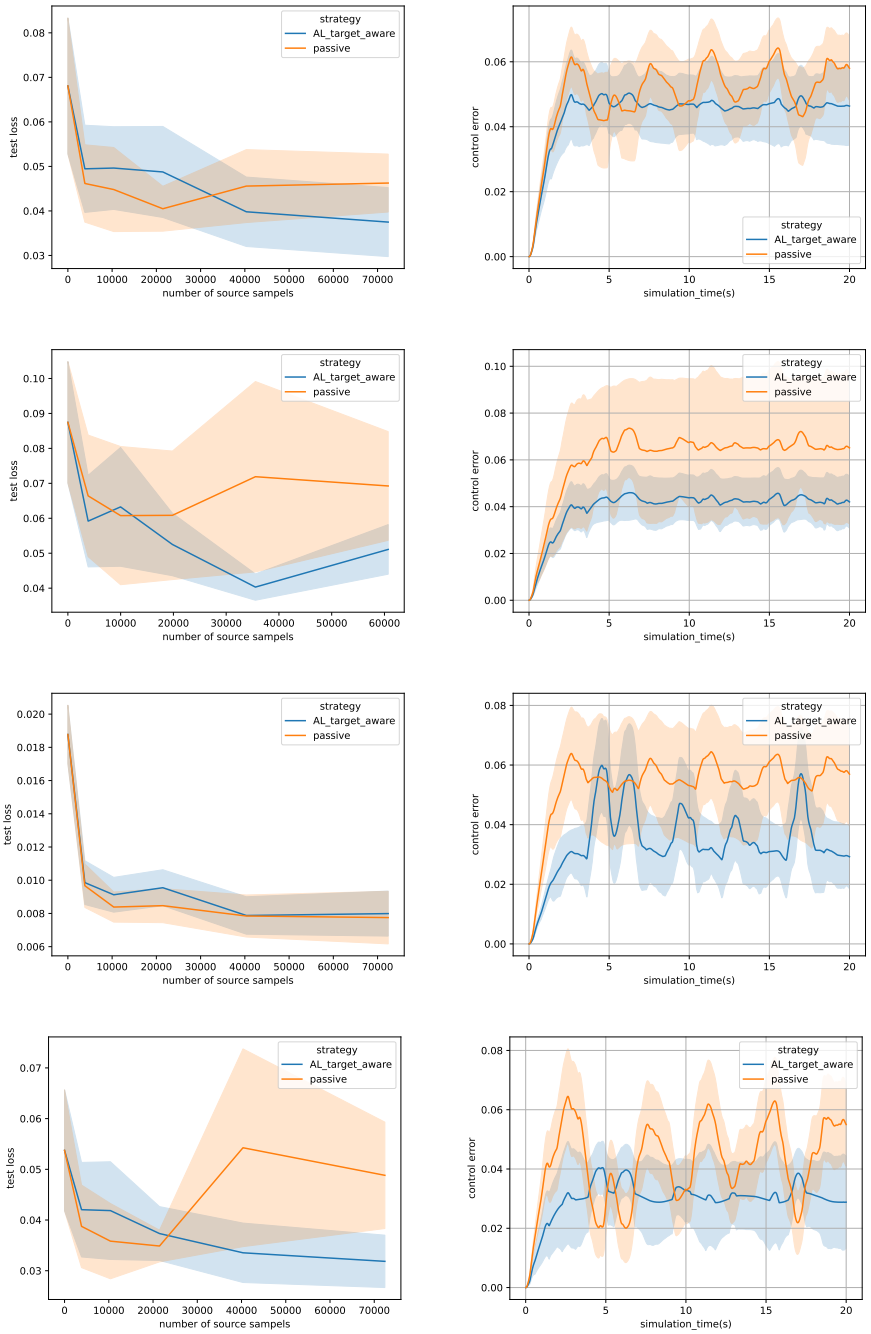

Figure 5: **Results on pendulum simulator for a specific target**. **Left:** The test loss of the estimated model $\hat{f}$. The passive strategy suffers from negative transfer while the active strategy steadily decreases. **Right:** The control error using final output $\hat{f}$. Here we use a model-based nonlinear policy $\pi(x, \hat{f})$. The model learned from active strategy leads to better control performance. From top to bottom, we have the unobservable $w_{\text{actual\_target}}$ as $[0, 0, 0.5, 0, 0.5, 0]$, $[0, 0, 1, 1, -1, 0]$, $[0, -1, 0.5, 0, 0.5, 0]$, $[0, 0.1, 0, -1, 0.5, 0]$. Overall, although AL does not always have a dominating advantage, most times it is more stable and can gain better test loss at the end.

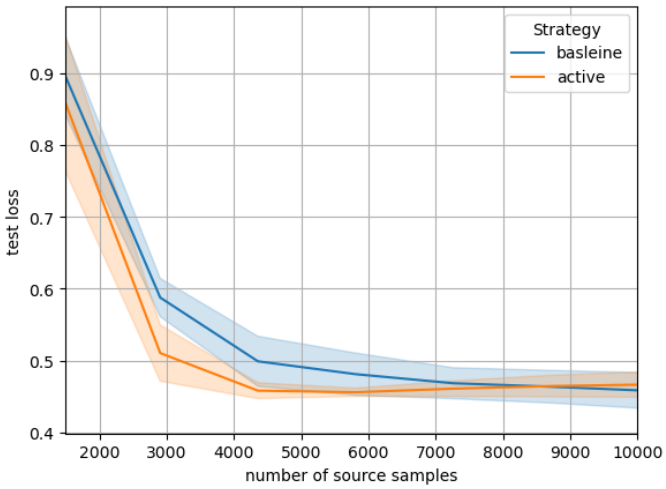

Figure 6: **Results on the real drone dataset** with target `drone_type_A_30_z` by using a neural net model. Our active strategy could converge faster than the passive strategy in the neural net model setting. Active strategy is able to converge faster than uniform sampling with smaller variances in the latter stage.

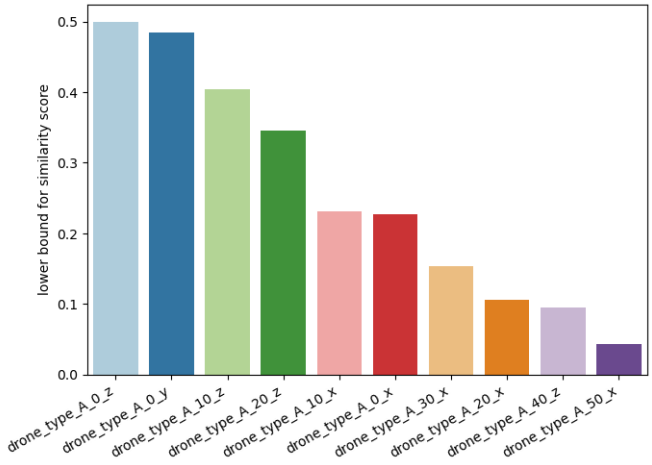

Figure 7: Top 10 the most similar source tasks. Again, given the target environment, the algorithm successfully finds the other `drone_type_A` environments as relevant sources, which aligns with our observation in the main paper.