Theoretical Foundations of Multi-task and Lifelong Learning

by

Anastasia Pentina

November, 2016

A thesis presented to the
Graduate School
of the
Institute of Science and Technology Austria, Klosterneuburg, Austria
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

The thesis of Anastasia Pentina, titled *Theoretical Foundations of Multi-task and Lifelong Learning*, is approved by:

Supervisor: Christoph Lampert, IST Austria, Klosterneuburg, Austria

Signature: _____

Committee Member: Jan Maas, IST Austria, Klosterneuburg, Austria

Signature: _____

Committee Member: Shai Ben-David, University of Waterloo, Waterloo, Canada

Signature: _____

Defense Chair: Beatriz Vicoso, IST Austria, Klosterneuburg, Austria

Signature: Beatriz Vicoso

I hereby declare that this thesis is my own work and that it does not contain other people's work without this being so stated; this thesis does not contain my previous work without this being stated, and the bibliography contains all the literature that I used in writing the dissertation.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other university or institution.

certify that any republication of materials presented in this thesis has been approved by the relevant publishers and co-authors.

Signature: 

Anastasia Pentina
November, 2016

Acknowledgments

First and foremost I would like to express my gratitude to my supervisor, Christoph Lampert. Thank you for your patience in teaching me all aspects of doing research (including English grammar), for your trust in my capabilities and endless support. Thank you for granting me freedom in my research and, at the same time, having time and helping me cope with the consequences whenever I needed it. Thank you for creating an excellent atmosphere in the group, it was a great pleasure and honor to be a part of it. There could not have been a better and more inspiring adviser and mentor.

I thank Shai Ben-David for welcoming me into his group at the University of Waterloo, for inspiring discussions and support. It was a great pleasure to work together. I am also thankful to Ruth Urner for hosting me at the Max-Planck Institute Tübingen, for the fruitful collaboration and for taking care of me during that not-so-sunny month of May.

I thank Jan Maas for kindly joining my thesis committee despite the short notice and providing me with insightful comments.

I would like to thank my colleagues for their support, entertaining conversations and endless table soccer games we shared together: Georg, Jan, Amelie and Emilie, Michal and Alex, Alex K. and Alex Z., Thomas, Sameh, Vlad, Mayu, Nathaniel, Silvester, Neel, Csaba, Vladimir, Morten. Thank you, Mabel and Ram, for the wonderful time we spent together. I am thankful to Shrinu and Samira for taking care of me during my stay at the University of Waterloo. Special thanks to Viktoriia for her never-ending optimism and for being so inspiring and supportive, especially at the beginning of my PhD journey.

Thanks to IST administration, in particular, Vlad and Elisabeth for shielding me from most of the bureaucratic paperwork.

I would like to thank my parents for being exceptionally motivating and supportive (even when it did not seem so) and for taking care of our dog when I was away. I would no be here without them.

My last, but not the least, words of gratitude are reserved for Vanya - it is thanks to you I was able to find myself and come to this point.

This dissertation would not have been possible without funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 308036.

Abstract

Traditionally machine learning has been focusing on the problem of solving a single task in isolation. While being quite well understood, this approach disregards an important aspect of human learning: when facing a new problem, humans are able to exploit knowledge acquired from previously learned tasks. Intuitively, access to several problems simultaneously or sequentially could also be advantageous for a machine learning system, especially if these tasks are closely related. Indeed, results of many empirical studies have provided justification for this intuition. However, theoretical justifications of this idea are rather limited.

The focus of this thesis is to expand the understanding of potential benefits of information transfer between several related learning problems. We provide theoretical analysis for three scenarios of multi-task learning - multiple kernel learning, sequential learning and active task selection. We also provide a PAC-Bayesian perspective on lifelong learning and investigate how the task generation process influences the generalization guarantees in this scenario. In addition, we show how some of the obtained theoretical results can be used to derive principled multi-task and lifelong learning algorithms and illustrate their performance on various synthetic and real-world datasets.

Table of Contents

1 Introduction

It has been a long-standing goal in machine learning to develop intelligent agents that would be able to persist in the world. Over the past decades significant progress has been made in developing efficient solutions for many related challenging problems, in particular in computer vision applications. Some of them even lead to super-human performance [25]. However, in order to succeed these methods often need an access to large amounts of annotated data. This is not a problem when the goal is to solve a particular isolated learning problem, due to the emergence of large data collections like ImageNet [79] with millions of images and hundreds of classes. However, it makes the application of many of the current machine learning methods for building intelligent systems doubtful, because it would require collecting extensive amounts of annotated data for every task the system faces during its life.

In contrast, humans are known to be able to learn new concepts from just a few examples. A plausible explanation of this gap is that machine learning problems are typically considered in isolation, while humans are able to exploit knowledge they acquired from previously learned tasks for solving new ones more efficiently. This observation has motivated an alternative, transfer approach to machine learning. It is based on the idea of transferring information between several related problems in order to improve the overall performance, instead of building a new model from scratch for every new learning task.

From the practical point of view, information transfer can be considered successful whenever, given the same amount of information, i.e. training data, it leads to better test performance than learning every task in isolation, as can be quantified by extensive empirical comparisons on various datasets. However, in this work we mainly focus on

the theoretical analysis of transfer learning. In particular, we are interested in identifying scenarios under which information transfer leads to provable reductions in the number of annotated examples needed for every considered task for obtaining reliable solutions.

In this thesis we focus on two learning scenarios - multi-task and lifelong learning. In the first case the learner is given a set of learning tasks simultaneously and its goal is to perform well on all of them. The success of a multi-task method depends on how the considered tasks are related and what information is transferred between them. In Chapter 3 we discuss three types of information transfer, particularly focusing on the use of linear predictors. In Section 3.1 we describe representation transfer that is based on the assumption that the tasks share a beneficial, potentially low-dimensional representation. In particular, we show that if such representation is given by a kernel function in some fixed set of kernels, then it is possible to infer this kernel based on the training data for several tasks. Moreover, under mild conditions on the kernel set, the sample complexity overhead associated with learning a kernel vanishes as the number of tasks increases and converges to the same sample complexity per task, as if that beneficial kernel was given to the learner. These results were published in the paper "Multi-task and Lifelong Learning of Kernels" with Shai Ben-David at ATL 2015 [70]. In Section 3.2 we consider parameter transfer approach that is based on the idea that the model parameters for the tasks are close to each other. In particular, we describe a theoretically justified method that is based on this assumption. This method processes tasks sequentially by transferring information between subsequent tasks and determines a beneficial task order based only on the training data for the given tasks. This is a joint work with Viktoriia Sharmanska and Christoph Lampert and was published in "Curriculum Learning of Multiple Tasks" at CVPR 2015 [74]. Finally, we also discuss what can be done if not for all tasks any annotated data is available in Section 3.3 [73].

In Chapter 4 we focus on the second, lifelong learning scenario, when the learner faces a stream of tasks and aims at extracting useful information from the observed tasks in order to perform well in the future on new ones. For this form of information transfer to make sense one has to make assumptions not only on the relatedness between the tasks, but also on the way they are ordered. We start with analyzing Baxter's model of lifelong learning [14], in which the tasks are assumed to be sampled i.i.d. from some unknown distributions. First, we show how the results obtained for

multi-task learning of kernels (Section 3.2) can be extended to lifelong learning. Then in Section 4.1.1 we describe a general PAC-Bayesian framework for lifelong learning and show how parameter and representation transfer approaches can be obtained from it. This is a joint work with Christoph Lampert published in "A PAC-Bayesian Bound for Lifelong Learning" at ICML 2014 [71]. After that, in Section 4.1.2 we investigate whether the i.i.d. assumption can be relaxed. First, we consider the case when the observed tasks are identically, but not independently distributed and provide a generalization of the PAC-Bayesian bound presented in Section 4.1.1. Next, we further relax the i.i.d. assumption by allowing the distribution generating the tasks to change over time. We show that in this case, under suitable assumptions, it is possible to learn a transfer procedure that is able to cope with the changes in the task generating distribution. This section is based on the joint work with Christoph Lampert, presented in "Lifelong Learning with Non-i.i.d. Tasks" at NIPS 2015 [72]. We conclude by discussing what can be done without any distributional assumptions on the tasks in Section 4.2. These results are presented in the joint work with Ruth Urner, "Lifelong Learning with Weighted Majority Votes" at NIPS 2016 [75].

2 Background

The focus of this work is to examine potential benefits of multi-task and lifelong learning compared to solving each task in isolation. To do so from the theoretical perspective we will utilize tools from statistical learning theory and compare the results to those established for solving a single task. Thus the purpose of this chapter is to provide the main notation that will be used in the manuscript as well as an overview of PAC (Probably Approximately Correct) [96] and PAC-Bayesian [62] theories that provide a formalism to analyze and compare machine learning methods in a principled way. For more details on this topic see [86, 66, 20].

2.1 Basic notions

Imagine we would like to write a spam filter - a computer program that, given an email, predicts whether it is a spam or not. One could try to hard-code all possible features that make an email a spam, but taking into account the amount of variation and ambiguity in the data this approach will likely fail. Instead one could design a system that, given a large corpora of emails that are spam and those that are not spam, would learn the distinctions. This type of approaches is studied in machine learning.

To design such a system one needs to define how objects of interest are represented to the computer. For example, one could represent emails as bag-of-words. This defines the space $\mathcal{X}$ of all possible objects (emails) which we will refer to as *input* or *feature space*. One also needs to specify what kind of predictions the system should make, i.e. the set of possible labels $\mathcal{Y}$. For the a spam filter $\mathcal{Y} = \{\text{"spam"}, \text{"not spam"}\}$. Finally, in

order to quantify the quality of the predictions, we will employ the notion of *loss function* $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ that for every pair of labels (y, y') specifies a penalty for predicting y instead of y'. In binary classification the typical choice is $0/1$ loss: $\ell(y, y') = [\![y \neq y']\!]$.

Now we can formally define the problem: given a *training set* $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of object-label pairs a learning system needs to output a *predictor* or *hypothesis* $h : \mathcal{X} \to \mathcal{Y}$ that maps objects to their labels. Of course, for a spam filter to be useful it should make as few mistakes as possible in the future on potentially new, yet unseen emails. However, for this to be possible the new, *test* instances should be related to what the system was trained one. For example, if for training the system was only given a collection of emails in English it is unreasonable to assume that it will be able to correctly identify spam emails in French. This is formalized in statistical learning theory through the notion of *task* D as a probability distribution over $\mathcal{X} \times \mathcal{Y}$. In particular, it is assumed that the object-label pairs in the training set S are sampled i.i.d. from some unknown task D and that the output of the learning system is tested on random examples coming from the same distribution, which is formally captured by the notion of *expected error*:

$$\mathrm{er}_D(h) = \mathop{\mathbb{E}}_{(x,y)\sim D} \left[\ell(h(x), y)\right], \tag{2.1}$$

In this work we mostly focus on *proper* learning, where the learner is required to output a hypothesis from a predefined *hypothesis class* $\mathcal{H}$. In this case the goal of the learner is to find a hypothesis with expected error close to the minimum value it can be, which is called *approximation error*:

$$\mathrm{opt}_D(\mathcal{H}) := \inf_{h\in\mathcal{H}} \mathrm{er}_D(h). \tag{2.2}$$

2.2 PAC learning

Since instead of the task distribution D the learner is given only a finite set of annotated examples S, it is not realistic to expect the learner to always find a good hypothesis. Therefore, instead, we are interested in algorithms that output a hypothesis with small expected error only with high probability. On top of that, the output of the learner is only needed to be a good approximation of the best hypothesis in the class. These

requirements are formally captured by the following definition, which is due to [35]:

Definition 1. (Agnostic PAC learnability) A hypothesis class $\mathcal{H}$ is *agnostic PAC learnable* if there exist a function $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm, such that for every $\epsilon, \delta \in (0,1)$, for every data distribution D over $\mathcal{X} \times \mathcal{Y}$ and every $n \geq n_{\mathcal{H}}(\epsilon, \delta)$ with probability at least $1-\delta$ over a training set S of size n:

$$\mathrm{er}_D(h) \leq \mathrm{opt}_D(\mathcal{H}) + \epsilon, \tag{2.3}$$

where $h \in \mathcal{H}$ is the hypothesis returned by the algorithm. The quantity $n_{\mathcal{H}}(\epsilon, \delta)$ is called *sample complexity* of class $\mathcal{H}$.

Learnability is closely related to the concept of uniform convergence.

Definition 2. (Uniform convergence) A hypothesis class $\mathcal{H}$ has *the uniform convergence* property if there exists a function $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$, for every data distribution D over $\mathcal{X} \times \mathcal{Y}$ and every $n \geq n_{\mathcal{H}}(\epsilon, \delta)$ with probability at least $1-\delta$ over a training set S of size n:

$$\forall h \in \mathcal{H} \,|\, \mathrm{er}_D(h) - \widehat{\mathrm{er}}_S(h)| \leq \epsilon, \tag{2.4}$$

where $\widehat{\mathrm{er}}_S(h)$ denotes the *empirical error*:

$$\widehat{\mathrm{er}}_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i). \tag{2.5}$$

If follows immediately from the above definitions that uniform convergence implies learnability:

$$\forall h \in \mathcal{H} : \mathrm{er}_D(h_S) \leq \widehat{\mathrm{er}}_S(h_S) + \epsilon \leq \widehat{\mathrm{er}}_S(h) + \epsilon \leq \mathrm{er}_D(h) + 2\epsilon, \tag{2.6}$$

where $h_S \in \arg\min_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h)$ is the *empirical risk minimizer (ERM)*. Moreover, in the case of binary classification, $\mathcal{Y} = \{-1, 1\}$, with $0/1$ loss, $\ell(y, y') = [\![y \neq y']\!]$, these two conditions are equivalent (Theorem 6.7 in [86]).

The uniform convergence property provides a way of proving learnability using techniques from probability theory. For a fixed hypothesis h the deviation between the expected error $\mathrm{er}_D(h)$ and its empirical counterpart $\widehat{\mathrm{er}}_S(h)$ can be bounded with high probability using concentration inequalities, like Hoeffding's [38]. However, in order to obtain a uniform convergence result one has to take into account the capacity of the hypothesis set $\mathcal{H}$. If $\mathcal{H}$ is finite, the number of its elements is a natural choice. However, even some of infinetly large hypothesis sets have the uniform convergence property. In the case of binary classification they are exactly those with finite VC dimension [97]:

Definition 3. (VC dimension) The *VC-dimension* of a binary hypothesis class $\mathcal{H}$, denoted by $\mathrm{VC}(\mathcal{H})$, is the maximal size of a set $C \subset \mathcal{X}$ such that the restriction of $\mathcal{H}$ to C is a set of all possible $2^{|C|}$ functions from C to $\{-1, 1\}$.

VC-dimension of a hypothesis class determines the number of samples needed to guarantee low estimation error:

Theorem 1 (Corollary 3.4 in [66])**.** *Let $\mathcal{H}$ be a family of functions taking values in $\{-1, 1\}$ with a finite VC-dimension and ℓ be the $0/1$ loss. Let S be a set of n training examples sampled i.i.d. from an unknown distribution D over $\mathcal{X} \times \{-1, 1\}$. Then for any $\delta > 0$ the following holds with probability at least $1 - \delta$ over the training set S for all $h \in \mathcal{H}$:*

$$\mathrm{er}_D(h) \leq \widehat{\mathrm{er}}_S(h) + \sqrt{\frac{2\,\mathrm{VC}(\mathcal{H})\log(en/\,\mathrm{VC}(\mathcal{H}))}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}}. \tag{2.7}$$

This result shows that the sample complexity of learning can be upper bounded by $\tilde{O}\left(\frac{\mathrm{VC}(\mathcal{H})+\log(1/\delta)}{\epsilon^2}\right)$[1]. However, a better bound of form $\tilde{O}\left(\frac{\mathrm{VC}(H)+\log(1/\delta)}{\epsilon}\right)$ can be obtained in the *realizable* case, when $\mathrm{opt}_D(\mathcal{H}) = 0$:

Theorem 2 (Corollaries 5.2 and 5.3 in [20])**.** *Let $\mathcal{H}$ be a class of binary functions with a finite VC-dimension and ℓ be the $0/1$ loss.There exists a constant C, such that for any $\delta \in (0, 1)$ and any learning task D with probability at least $1 - \delta$ over a training set S of*

[1] We use $\tilde{O}$ to hide some of the logarithmic factors.

size n, sampled i.i.d. from D:

$$\mathrm{er}_D(h_S) \leq \widehat{\mathrm{er}}_S(h_S) + \sqrt{\widehat{\mathrm{er}}_S(h_S) \cdot \Delta} + \Delta, \tag{2.8}$$

$$\widehat{\mathrm{er}}_S(h_S) \leq \mathrm{er}_D(h_S) + \sqrt{\mathrm{er}_D(h_S) \cdot \Delta} + \Delta, \tag{2.9}$$

$$\mathrm{er}_D(h_S) \leq \mathrm{opt}_D(\mathcal{H}) + \sqrt{\mathrm{opt}_D(\mathcal{H}) \cdot \Delta} + \Delta, \tag{2.10}$$

where $h_S \in \arg\min_{h \in \mathcal{H}} \widehat{\mathrm{er}}_S(h)$ is an empirical risk minimizer and

$$\Delta = C \frac{\mathrm{VC}(\mathcal{H}) \log(n) + \log(1/\delta)}{n}. \tag{2.11}$$

Theorems 1 and 2 show that whenever the VC-dimension of a hypothesis class is small, an empirical risk minimizer is guaranteed to work well. However, requiring a small VC-dimension seriously limits the approximation properties of the class. In particular, it may lead to large $\mathrm{opt}_D(\mathcal{H})$. A possible way to overcome this problem is to modify the functional to be minimized based on the training data.

Consider the class of linear classifiers:

$$\mathcal{H} = \{h_w : x \mapsto \mathrm{sign}(\langle w, x \rangle) \mid w \in \mathbb{R}^d, \|w\| \leq 1\}. \tag{2.12}$$

Note that the constraint on the norm of the weight vector w does not influence the capacity of the hypothesis class, because only the sign of the output is used for making predictions. The VC-dimension of this class is d and therefore the sample complexity of ERM is $\tilde{O}(d/\epsilon^2)$, which can be problematic for high-dimensional problems. However, if instead of searching for a hypothesis that just makes few mistakes on the training set, the learner would prefer classifiers that also produce confident predictions, the sample complexity can be reduced. This idea is captured by the notion of *margin error*:

$$\mathrm{er}_D^{\gamma}(h_w) := \mathop{\mathbb{E}}_{(x,y)\sim D} [\![y\langle w, x \rangle < \gamma]\!]. \tag{2.13}$$

Note that $\mathrm{er}_D(h) \leq \mathrm{er}_D^{\gamma}(h)$ for every $\gamma \geq 0$. The following theorem provides a bound on the sample complexity of minimizing the margin loss (a similar bound in the opposite direction of the same form can also be obtained):

Theorem 3 (Corollary 4.1 in [66])**.** *Let $\mathcal{H}$ be a set of linear predictors and assume that $\mathcal{X} \subset \{x \in \mathbb{R}^d : \|x\| \leq B\}$. Fix $\gamma > 0$. Then for any $\delta > 0$ with probability at least $1 - \delta$ the following holds for any $h_w \in \mathcal{H}$:*

$$\mathrm{er}_D(h_w) \leq \widehat{\mathrm{er}}_S^{\gamma}(h_w) + 2\sqrt{\frac{B^2}{\gamma^2 n}} + \sqrt{\frac{\log(1/\delta)}{2n}}, \tag{2.14}$$

where $\widehat{\mathrm{er}}_S^{\gamma}$ is the empirical margin loss:

$$\widehat{\mathrm{er}}_S^{\gamma}(h_w) = \frac{1}{n}\sum_{i=1}^{n} [\![y_i \langle w, x \rangle < \gamma]\!]. \tag{2.15}$$

The most important property of the above bound is that it does not depend on the dimensionality d. In particular, it holds even for infinitely dimensional spaces. This may seem to contradict the fact that infinitely dimensional linear predictors have infinite VC-dimension and therefore are not learnable. However, this is not the case because Theorem 3 is meaningful only if there exists a predictor with low margin error, which is a significantly stronger requirement than that of having a small expected error with $0/1$ loss.

Theorem 3 suggests that one could enrich the expressive power of linear predictors by first mapping the data into a high dimensional space and then learn a halfspace there. This idea motivates more general *kernel* approaches.

Definition 4. (Kernel function) A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *kernel*, if there exist a Hilbert space F and a mapping $\phi : \mathcal{X} \to F$ with inner product $\langle \cdot, \cdot \rangle$ such that $K(x, x') = \langle \phi(x), \phi(x') \rangle$ for all $x, x' \in \mathcal{X}$.

A kernel function allows treating potentially non-linear predictors on $\mathcal{X}$ as linear, only in a different space F:

$$\mathcal{H}_K = \{h_w : x \mapsto \mathrm{sign}(\langle w, \phi(x) \rangle) \mid \|w\|_K \leq 1\}. \tag{2.16}$$

Moreover, many learning algorithms for halfspaces can be performed only based on the values of the kernel function for pairs of training examples, so there is no need to specify

the mapping ϕ explicitly. In addition, it makes them more computationally tractable than performing a straightforward empirical risk minimization.

Theorem 3 applies to $\mathcal{H}_K$ with just a modification of $B^2 = \sup_{x \in \mathcal{X}} K(x, x)$.

2.3 PAC-Bayesian learning

In contrast to PAC learning that deals with deterministic predictors, PAC-Bayesian theory analyses randomized, *Gibbs* predictors. Given a distribution Q over the hypothesis set $\mathcal{H}$, the corresponding Gibbs predictor for every input $x \in \mathcal{X}$ samples a hypothesis h according to Q and returns its prediction $h(x)$. Its expected error on a task D can then be written as:

$$\mathrm{er}_D(Q) = \mathop{\mathbb{E}}_{h \sim Q} \mathop{\mathbb{E}}_{(x,y) \sim D} \ell(h(x), y) \tag{2.17}$$

and its empirical counter-part computed based on a training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is:

$$\widehat{\mathrm{er}}_S(Q) = \mathop{\mathbb{E}}_{h \sim Q} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i). \tag{2.18}$$

PAC-Bayesian bounds allow us to relate these two quantities and also explicitly incorporate prior knowledge in a form of some *prior* distribution P over the hypothesis set $\mathcal{H}$. There are many ways of proving PAC-Bayesian bounds that lead to slightly different results [22, 84]. In this work we will concentrate of the following, probably the simplest form:

Theorem 4. *Let ℓ be a loss functions that takes values in $[0, 1]$ and P be any prior distribution over the hypothesis set $\mathcal{H}$ that should be selected before seeing the training data. Then for any $\delta > 0$ with probability at least $1 - \delta$ over a training set S of size n for all distributions Q over $\mathcal{H}$ the following holds:*

$$\mathrm{er}_D(Q) \leq \widehat{\mathrm{er}}_S(Q) + \frac{1}{\sqrt{n}} \left(\mathrm{KL}(Q||P) + \frac{1}{8} + \log \frac{1}{\delta} \right), \tag{2.19}$$

where

$$\mathrm{KL}(Q||P) = \mathop{\mathbb{E}}_{h \sim Q} \ln \frac{Q(h)}{P(h)} \tag{2.20}$$

is the Kullback-Leibler divergence.

Proof. The main ingredient of PAC-Bayesian bounds is the following inequality [28] that holds for any $\lambda > 0$:

$$\mathop{\mathbb{E}}_{h\sim Q} g(h) \leq \frac{1}{\lambda}\left(\mathrm{KL}(Q||P) + \log \mathop{\mathbb{E}}_{h\sim P} e^{\lambda g(h)}\right). \tag{2.21}$$

By applying the above inequality to $g(h) = n(\mathrm{er}_D(h) - \widehat{\mathrm{er}}_S(h))$ we obtain:

$$n(\mathrm{er}_D(Q) - \widehat{\mathrm{er}}_S(Q)) \leq \frac{1}{\lambda}\left(\mathrm{KL}(Q||P) + \log \mathop{\mathbb{E}}_{h\sim P} e^{\lambda n(\mathrm{er}_D(h) - \widehat{\mathrm{er}}_S(h))}\right). \tag{2.22}$$

Note that:

$$e^{\lambda g(h)} = \prod_{i=1}^{n} \exp(\lambda(\mathrm{er}_D(h) - \ell(h(x_i), y_i))). \tag{2.23}$$

Since for any fixed $h \in \mathcal{H}$ the factors are independent, by applying Hoeffding's lemma one obtains that:

$$\mathop{\mathbb{E}}_{S\sim D^n} e^{\lambda g(h)} \leq e^{\frac{\lambda^2 n}{8}}. \tag{2.24}$$

Using the fact that the prior P does not depend on the training data S we get that:

$$\mathop{\mathbb{E}}_{S\sim D^n} \mathop{\mathbb{E}}_{h\sim P} e^{\lambda g(h)} \leq e^{\frac{\lambda^2 n}{8}}. \tag{2.25}$$

Therefore, by Markov's inequality with probability at least $1-\delta$:

$$\log \mathop{\mathbb{E}}_{h\sim P} e^{\lambda g(h)} \leq \frac{\lambda^2 n}{8} + \log(1/\delta). \tag{2.26}$$

By choosing $\lambda = 1/\sqrt{n}$ we obtain the statement of the theorem. □

In terms of technicality the above proof is much more basic than those needed for proving bounds like Theorems 1 and 2. This is because randomized predictors allow substituting the worst case analysis, i.e. union bounds employed in PAC analysis, and the corresponding combinatorial arguments by the averages and Markov's inequality. Consequently, the above theorem does not involve any combinatorial measure of capacity of the hypothesis set $\mathcal{H}$. Instead, it contains the Kullback-Leibler divergence between the prior and the posterior distributions, which makes it data-dependent. In particular, it suggests that in order to guarantee a small upper bound on the expected

loss one should try to search for a posterior distribution that leads to small empirical error, but at the same time is not significantly different from the prior, which is captured by KL divergence and can be seen as some form of regularization. Consequently, the minimization of the right hand side of (2.19) does not lead to just empirical risk minimization, as it is for PAC bounds, but allows to develop other algorithms that can also take into account the prior knowledge about the problem. And the fact that the bound holds uniformly for all posterior distributions means that its guarantee will also hold for the one minimizing its right hand side.

On the other hand, the guarantees provided by Theorem 4 are for Gibbs predictors, which are rarely used in practice. However, there are situations when equation (2.19) can be converted into one about standard, deterministic predictors. In particular, in case of binary classification with $0/1$ loss, the expected error of a majority vote classifier, corresponding to a distribution Q, is at most twice $\mathrm{er}_D(Q)$ [63, 51]:

$$\mathop{\mathbb{E}}_{(x,y)\sim D} \left[\!\!\left[\mathrm{sign}\left(\mathop{\mathbb{E}}_{h\sim Q} h(x) \right) \neq y \right]\!\!\right] \leq 2\,\mathrm{er}_D(Q). \tag{2.27}$$

This is because, for a fixed point (x, y), the majority will make a mistake on it if and only if at least a half of the predictors, with respect to distribution Q, are also giving an incorrect prediction on it, in which case $\mathbb{E}_{h\sim Q}[\![h(x) \neq y]\!] \geq 0.5$.

A concrete form of an objective function provided by the right hand side of (2.19) depends on the choice of the hypothesis set $\mathcal{H}$ and a form of prior and posterior distributions. The best understood case are linear predictors and Gaussian distributions. In particular, let $P = \mathcal{N}(w_P, I_d)$ and $Q = \mathcal{N}(w_Q, I_d)$, i.e. Gaussian distributions with unit variance that differ only by the value of their means. First, one has to compute the empirical error of a Gibbs classifier. For this choice of posterior distributions it is [33, 50]:

$$\widehat{\mathrm{er}}_S(Q) = \frac{1}{n} \sum_{i=1}^{n} \bar{\Phi}\left(\frac{y_i \langle w_Q, x_i \rangle}{\|x_i\|} \right), \tag{2.28}$$

where

$$\bar{\Phi}(z) = \frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{z}{\sqrt{2}} \right) \right) \tag{2.29}$$

and

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\phi}} \int_0^z e^{-t^2} dt. \tag{2.30}$$

The Kullback-Leibler divergence between $\mathcal{N}(w_P, I_d)$ and $\mathcal{N}(w_Q, I_d)$ is simply $\frac{\|w_P - w_Q\|^2}{2}$. Lastly, note that the majority vote classifier associated with $\mathcal{N}(w_Q, I_d)$ is identical to the one given by its mean, i.e. w_Q. Therefore the bound (2.19) leads to the following result for linear predictors:

$$\frac{1}{2} \operatorname{er}_D(h_{w_Q}) \leq \frac{1}{n} \sum_{i=1}^{n} \bar{\Phi} \left(\frac{y_i \langle w_Q, x_i \rangle}{\|x_i\|} \right) + \frac{\|w_P - w_Q\|^2}{2\sqrt{n}} + \frac{\log 1/\delta + 1/8}{\sqrt{n}}. \tag{2.31}$$

Now we can see that if we set w_P to be the zero vector, we obtain an objective function similar to one used in Support Vector Machines (SVM).

3 Multi-task Learning

In multi-task learning [21, 2] a learner is given a collection $D_1, \ldots, D_T$ of T prediction tasks that need to be solved. As most of the previous works, we will assume that all of them are defined on the same domain $\mathcal{X} \times \mathcal{Y}$, however, there were some attempts to generalize multi-task learning to heterogeneous representations [100]. For each task D_t the learner is given a training set of i.i.d. annotated examples $S_t = \{(x_1^t, y_1^t), \ldots, (x_n^t, y_n^t)\}$. We will also assume that the size of the training sets n is the same for all tasks. Given a hypothesis set $\mathcal{H}$ and a loss function ℓ the goal of the learner is to identify a predictor h_t for every task D_t, such that the resulting *average* expected error is small:

$$\mathrm{er}_{\mathbf{D}}(\mathbf{h}) = \frac{1}{T} \sum_{t=1}^{T} \mathrm{er}_{D_t}(h_t), \tag{3.1}$$

where $\mathbf{D} = (D_1, \ldots, D_T)$ and $\mathbf{h} = (h_1, \ldots, h_T)$.

Traditional machine learning algorithms can be applied to solve each of the given tasks in isolation. However, the motivation behind the multi-task scenario is that by solving all given tasks jointly and transferring information between them the learning process can be made more efficient. The success of information transfer depends on how the given tasks are related. Various assumptions have been exploited in the multi-task literature, ranging from Bayesian approaches [10, 36], to learning shared metrics for nearest neighbor classifiers [92], shared low-dimensional representations [6] or joint regularization [32] for linear classification.

There are also numerous works aiming at demonstrating potential benefits of information transfer from the theoretical perspective through sample complexity reductions. One of the first and most general analysis of multi-task learning was performed by

Baxter [13, 14]. Assuming that instead of considering a single hypothesis set $\mathcal{H}$, the learner is given a collection of hypothesis classes $\mathbb{H}$, Baxter proved uniform convergence bounds of the following form (Theorem 4 in [14]) for all $\mathbf{h} \in \mathbb{H}^T = \cup_{\mathcal{H}\in\mathbb{H}}\{(h_1, \ldots, h_T) : h_1, \ldots, h_T \in \mathcal{H}\}$:

$$\mathrm{er}_{\mathbf{D}}(\mathbf{h}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{h}) + \epsilon_{mt}, \tag{3.2}$$

where

$$\widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{h}) = \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathrm{er}}_{S_t}(h_t) \tag{3.3}$$

is an average empirical error evaluated on the collection of training sets $\mathbf{S} = \{S_1, \ldots, S_T\}$. Alternatively one could simply sum up individual uniform convergence bounds for T tasks using the union bound argument and obtain a similar result, but for all $\mathbf{h} \in \mathbb{H}^T_{ind} = \{(h_1, \ldots, h_T) : \ h_1, \ldots, h_T \in \cup_{\mathcal{H}\in\mathbb{H}}\mathcal{H}\}$:

$$\mathrm{er}_{\mathbf{D}}(\mathbf{h}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{h}) + \epsilon_{ind}. \tag{3.4}$$

On one hand, the second approach gives a more general statement, because considers a larger hypothesis class: $\mathbb{H}^T \subset \mathbb{H}^T_{ind}$. On the other hand, ϵ_{ind} in (3.4) decreases only with n (as, for example, in Theorem 1), while ϵ_{mt}, as shown in [14], under some circumstances decreases with both n and T and can potentially be smaller. Therefore, in general (3.2) and (3.4) are not comparable. However, if we assume that tasks are related in a way that there exists $\mathcal{H} \in \mathbb{H}$ that works well for all T given tasks, i.e. the empirical error on the right hand side of (3.2) can be made as small as that in (3.4), and the number of tasks T is large, Baxter's result shows that the number of training examples per task needed to achieve particular estimation error is smaller than that given by considering every task in isolation, as in (3.4). This is one of the examples of how relatedness between tasks may lead to provable sample complexity reductions in comparison to solving each given task independently from the others.

Baxter's analysis was later improved in [61] by using Rademacher complexities. Similar results were also obtained in [18] under the assumption that the task distributions can be transformed one to another by applying transformation functions from a predefined set of possible transformations.

In this chapter we will discuss some of the possible relatedness assumptions in more

details, including representation transfer and hypothesis transfer, and illustrate potential benefits of joint learning from both theoretical and experimental perspectives.

3.1 Representation transfer

One of the widely used assumptions on task relatedness is that there exists a common representation that leads to low average approximation error for all tasks. In particular, this assumption was explored in case where every task is solved using a sparse combination of original features or their linear transformations [5, 6]. Assuming that the sparsity pattern is shared across the given learning problems, the corresponding methods aim at inferring it from the data for all tasks. In its simplest form, where only sparse combinations of the original features are considered, it is achieved through minimizing the following objective function:

$$\widehat{\mathrm{er}}_{\mathbf{S}}(h_{w_1}, \ldots, h_{w_T}) + \gamma \|W\|_{2,1}^2, \tag{3.5}$$

where $h_{w_t}(x) = \langle w_t, x \rangle$ are linear predictors and W is a matrix with columns given by the weight vectors $w_1, \ldots, w_T$. First, the objective function (3.5) favors linear predictors that lead to small average empirical error (3.3). Moreover, the regularization term $\|W\|_{2,1}^2 = \left(\sum_k \|w^k\|_2\right)^2$, where w^k denotes the k-th row of matrix W, enforces the weight matrix W to have many rows equal to zero. Thus minimizing (3.5) leads to linear predictors that well perform on all tasks and share a common sparsity pattern.

These methods were further extended to be able to handle different levels of relatedness between tasks [8], disjoint [7] or overlapping [47] groups of related tasks and exploit known unrelated tasks [78]. A similar paradigm was also used in [1], where the predictors for the tasks are assumed to lie on a low-dimensional manifold, rather than in a linear subspace. Furthermore, it was extended to kernel methods, where the common representation is assumed to be described by a kernel function and the corresponding methods aim at discovering a suitable kernel [39, 40, 34, 83, 76, 101].

The assumption of a common low-dimensional representation can be seen as a particular case of Baxter's model where tasks share a good hypothesis space. Therefore his analysis [13, 14] can be directly applied to this scenario. However, potential sample

complexity improvements provided by these results depend on the behavior of particular types of covering numbers, which, due to generality of Baxter's results, is often not easy to infer. This motivated a series of works analyzing specifically possible benefits of inferring a low-dimensional representation from a group of tasks, either as a sparse combination of initial features [58, 45, 53] or their linear transformations [55, 59, 60]. In this section we extend these results by providing an analysis for learning a common kernel function for multiple tasks. There results have been published in the paper "Multi-task and Lifelong Learning of Kernels" [70].

Multiple kernel learning for single-task problems has been studied theoretically using various techniques. Cortes et al in [26] have analyzed the case of linear combinations of finitely many kernels with ℓ_p constraints using Rademacher complexity. In particular, for ℓ_1 constraint they have provided a bound of form $O(\sqrt{\log(k)/n})$, where k is the number of base kernels and m is the size of the training set. This analysis was further improved using local Rademacher complexities in [44]. We will instead employ the technique from [89] that is based on the notion of pseudodimension:

Definition 5. The class $\mathcal{K} = \{K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}\}$ of kernels *pseudo-shatters* the set of n pairs of points $(x_1, x'_1), \ldots, (x_n, x'_n)$, if there exist thresholds $t_1, \ldots, t_n$ such that for any $b_1, \ldots, b_n \in \{-1, +1\}$ there exists a kernel $K \in \mathcal{K}$ such that $\operatorname{sign}(K(x_i, x'_i) - t_i) = b_i$. The *pseudodimension* $d_\phi(\mathcal{K})$ is the largest n such that there exists a set of n pairs pseudo-shattered by $\mathcal{K}$.

This technique leads to suboptimal dependence on the number of kernels in the case of linear combinations: $O(\sqrt{k/n})$ instead of $O(\sqrt{\log(k)/n})$ from [26]. However, it allows us to obtain general results that hold for any kernel family with finite pseudodimension. In particular, the following upper bounds on the pseudodimension were shown in [89]:

- convex or linear combinations of k kernels have $d_\phi \leq k$;
- Gaussian families with learned covariance matrix in $\mathbb{R}^k$ have $d_\phi \leq \frac{k(k+1)}{2}$;
- Gaussian families with learned low-rank covariance have $d_\phi \leq kr \log_2(8ekr)$, where r is the maximum rank of the covariance matrix.

In order to analyze multiple kernel learning in the multi-task setting we will use the following multi-task version of the margin loss:

$$\widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathbf{h}) = \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{er}}^{\gamma}_{S_t}(h_t) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}[\![y_i^t h_t(x_i^t) < \gamma]\!], \tag{3.6}$$

where $\mathbf{S} = \{S_1 \ldots, S_T\}$ is the collection of training sets for all tasks. By employing the same approach as in Theorem 3, we obtain the following uniform-convergence type of bound:

Theorem 5 (Theorem 3 in [70])**.** *Let $D_1, \ldots, D_T$ be any T learning tasks, defined on $\mathcal{X} \times \{-1, +1\}$, and $\mathcal{K}$ be any kernel family with finite pseudodimension, such that $K(x, x) \leq B^2$ for any $K \in \mathcal{K}$ and any $x \in \mathcal{X}$. Then, for any fixed $\gamma > 0$ and any $\delta > 0$, if $n > 2/\epsilon^2$, then, with probability at least $1 - \delta$ over the training set $\mathcal{S} = \{S_1, \ldots, S_T\}$, where $S_t \sim D_t^n$ for $t = 1, \ldots, T$, the following inequality holds for all $\mathbf{h} = (h_1, \ldots, h_T) \in \cup_{K\in\mathcal{K}}\mathcal{H}_K^T := \cup_{K\in\mathcal{K}}\{(h_1, \ldots, h_T) :\ h_1, \ldots, h_T \in \mathcal{H}_K\}$:*

$$\mathrm{er}^{2\gamma}_{\mathbf{D}}(\mathbf{h}) + \epsilon \ \geq\ \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathbf{h}) \ \geq\ \mathrm{er}_{\mathbf{D}}(\mathbf{h}) - \epsilon, \tag{3.7}$$

where

$$\mathrm{er}^{2\gamma}_{\mathbf{D}} = \frac{1}{T}\sum_{t=1}^{T}\mathrm{er}^{2\gamma}_{D_t}(h_t) = \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{(x,y)\sim D_t}[\![yh_t(x) < 2\gamma]\!] \qquad \textit{and} \tag{3.8}$$

$$\epsilon = \sqrt{8\frac{\frac{2\log 2 - \log\delta}{T} + \log 2 + \frac{d_\phi(\mathcal{K})}{T}\log\frac{128eT^2n^3B^2}{\gamma^2 d_\phi(\mathcal{K})} + \frac{256B^2}{\gamma^2}\log\frac{\gamma en}{8B}\log\frac{128nB^2}{\gamma^2}}{n}}. \tag{3.9}$$

Analogously to the case of uniform convergence in PAC learning, the above result leads to justification of the empirical risk minimization approach:

Corollary 1. *Let $\widehat{\mathbf{h}}$ be a minimizer, over $\cup_{K\in\mathcal{K}}\mathcal{H}_K^T$, of the empirical γ-margin loss, $\widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathbf{h})$. Then for any $\mathbf{h}^* \in \cup_{K\in\mathcal{K}}\mathcal{H}_K^T$ (and in particular for a minimizer over of the true 2γ-loss $\mathrm{er}^{2\gamma}_{\mathbf{D}}(\mathbf{h})$):*

$$\mathrm{er}_{\mathbf{D}}(\widehat{\mathbf{h}}) \leq \mathrm{er}^{2\gamma}_{\mathbf{D}}(\mathbf{h}^*) + 2\epsilon.$$

Proof. The result is implied by the following chain of inequalities:

$$\mathrm{er}_{\mathbf{D}}(\widehat{\mathbf{h}}) - \epsilon \leq_1 \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\widehat{\mathbf{h}}) \leq_2 \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathbf{h}^*) \leq_3 \mathrm{er}^{2\gamma}_{\mathbf{D}}(\mathbf{h}^*) + \epsilon$$

where $(\leq_1)$ and $(\leq_3)$ follow from the above theorem and $(\leq_2)$ follows from the definition of an empirical risk minimizer. □

Note that, while Theorem 5 provides performance guarantees for the ERM rule, in many cases implementing it is NP-hard as is implementing of ERM in the agnostic case for single-task problems [41].

Note, that in the case of a single task ($T = 1$) Theorem 5 leads the same bound of form $\tilde{O}\left(\sqrt{\frac{d_\phi + B^2/\gamma^2}{n}}\right)$ as the results of [89]. However, as the number of tasks (T) tends to infinity, while the number of training examples per task (n) stays constant, the overhead associated with learning a kernel vanishes. In particular, the bound on the estimation error turns into $\tilde{O}\left(\sqrt{\frac{B^2/\gamma^2}{n}}\right)$, i.e. the bound known for the case of learning with a single kernel (Theorem 3). Therefore, if there exists a kernel $K \in \mathcal{K}$ that is useful for all tasks, i.e. $\mathrm{er}^{2\gamma}_{\mathbf{D}}(\mathbf{h}^*)$ is small, then access to training data from sufficiently many tasks allows to learn them with the same sample complexity per task, as if that good kernel was known in advance.

3.2 Parameter transfer

The parameter transfer approach is based on the idea that predictors corresponding to related tasks are similar to each other in terms of their parametric form. In the case of linear predictors this idea was introduced in [32], where the authors proposed an SVM-like algorithm:

$$\begin{aligned} &\min_{w_0, w_t, \xi_i^t} \|w_0\|^2 + \frac{1}{T}\sum_{t=1}^{T} \|w_t - w_0\|^2 + \frac{C}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}\xi_i^t \\ &\text{subject to } \; y_i^t \langle w_t, x_i^t \rangle \geq 1 - \xi_i^t, \quad \xi_i^t \geq 0 \quad \text{for all } t, i. \end{aligned} \tag{3.10}$$

The regularization term in (3.10) enforces the weight vectors for different tasks to lie close to each other in terms of ℓ_2 norm and thus represents the parameter transfer

assumption. A similar idea was also used in [23] in combination with online perceptron algorithm and applied to various compluter vision problems [24, 90].

The main limitation of algorithm (3.10) is that it treats all tasks symmetrically as equally related. This property reduces its applicability in realistic scenarios, where there might be some outlier tasks or disjoint groups of related tasks and enforcing information transfer may cause decrease of prediction quality. In such cases more flexible models that are able to exploit the underlying task relatedness structure would be preferable. This can be achieved by using graph regularization [31]. However, it requires prior knowledge about the level of similarity between the tasks. Alternatively one could allow the algorithm to automatically determine and exploit the structure of task relations. In particular, we will discuss how to do it in a principled way in the case, when tasks are solved sequentially, one at a time by transferring information from previous tasks to the current one. These results were published in the paper "Curriculum Learning of Multiple Tasks" [74].

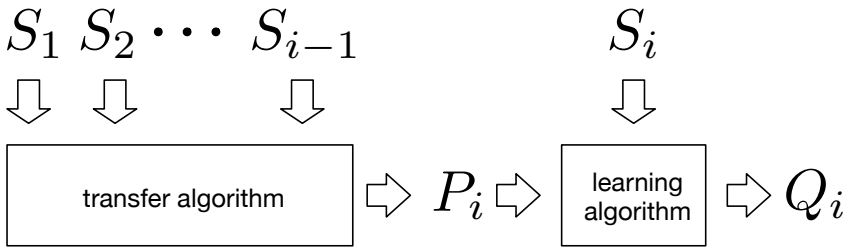


Figure 3.1: Illustration of sequential multi-task learning.

We use the PAC-Bayesian approach to analyze this setting. For every task t let Q_t and P_t denote the posterior and prior distributions over $\mathcal{H}$. We assume that there is some deterministic transfer algorithm $\mathcal{TA}$ that produces a prior for the current task based on the previous tasks. In addition, there is a learning algorithm $\mathcal{A}$ that solves the task based on this prior knowledge (see Figure 3.1). In contrast to (3.10), this approach provides flexibility in the sense that not all the tasks need to be equally related. However, one would expect that its effectiveness depends on the chosen task order. The following generalization bound allows us to quantify these effects.

Theorem 6 ([74])**.** *For any deterministic transfer algorithm $\mathcal{TA}$, any deterministic learning algorithm $\mathcal{A}$, any prior distribution P and any $\delta > 0$, the following inequality holds with probability at least $1-\delta$ over the training sets $S_1, \ldots, S_T$ of size n each for all orders*

$\pi \in \mathcal{S}_T$ *of* T *tasks:*

$$\mathrm{er}_{\mathbf{D}}(\mathbf{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{Q}) + \frac{1}{T\sqrt{n}} \sum_{t=1}^{T} \mathrm{KL}(Q_{\pi(t)} || P_{\pi(t)}) + \frac{8 \log T + 1}{8\sqrt{n}} + \frac{\log 1/\delta}{T\sqrt{n}}, \tag{3.11}$$

where

$$\mathbf{Q} = (Q_1, \ldots, Q_T) \tag{3.12}$$

$$Q_{\pi(t)} = \mathcal{A}(S_{\pi(t)}, P_{\pi(t)}) \tag{3.13}$$

$$P_{\pi(t)} = \begin{cases} P \ \ \textit{for} \ t = 1 \\ \mathcal{TA}(S_{\pi(1)}, \ldots, S_{\pi(t-1)}) \ \ \textit{for} \ t \geq 2 \end{cases} \tag{3.14}$$

$$\mathrm{er}_{\mathbf{D}}(\mathbf{Q}) = \frac{1}{T} \sum_{t=1}^{T} \mathrm{er}_{D_t}(Q_t) = \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{h \sim Q_t} \mathop{\mathbb{E}}_{(x,y) \sim D_t} \ell(h(x), y) \tag{3.15}$$

$$\widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{Q}) = \frac{1}{T} \sum_{t=1}^{T} \widehat{\mathrm{er}}_{S_t}(Q_i) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{h \sim Q_t} \ell(h(x_i^t), y_i^t) \tag{3.16}$$

The above theorem provides an upper bound on the average expected error over all tasks - the quantity of interest that the learner would like to minimize but cannot compute directly. In contrast, the right hand side of (3.11) consists only of computable quantities: the average empirical error and a sum of Kullback-Leibher divergences between priors and posteriors for every task. Moreover, it also depends on the order π, in which tasks are processed, because the prior for every task $\pi(t)$ depends on the tasks $\pi(1), \ldots, \pi(t-1)$ that were solved before. Therefore the right hand side of (3.11) can be seen as a quality measure of the order π and by minimizing it one can obtain an order that is well suited for solving the tasks based on the given training data. In addition, because the inequality holds uniformly for all possible task orders, its guarantees will also holds for the resulting, learned data-dependent order.

We illustrate this process for the case of linear binary classification. We assume that $\mathcal{X}$ is a subset of $\mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$, ℓ is the zero-one loss and $\mathcal{H}$ is the set of all linear classifiers without a bias term $\{\mathrm{sign}\langle w, x\rangle)\}$, where $w \in \mathbb{R}^d$ is a weight vector. To capture this setting in the PAC-Bayesian framework we set all prior and posterior distributions to

be Gaussian with unit variance that differ only by the value of their means:

$$Q_t = \mathcal{N}(w_Q^t, I_d), \qquad P_t = \mathcal{N}(w_P^t, I_d) \qquad \text{for } t = 1, \ldots, T. \tag{3.17}$$

In order to apply Theorem 6 one also needs to specify the transfer and the learning algorithms. We consider a simple transfer algorithm that just remembers the solution for the last solved task. Thus the prior for the current task is equal to the posterior obtained for the previous one: $P_{\pi(t)} = Q_{\pi(t-1)}$. We also set the initial prior P to be the standard normal distribution $\mathcal{N}(\mathbf{0}, I_d)$ representing the absence of prior knowledge at the beginning of the learning process. For the learning algorithm $\mathcal{A}$ we select a widely used in computer vision applications Adaptive SVM [42], which is a modification of the SVM with biased regularization. For a given weight vector $\tilde{w}$ and training data for a task, it performs the following optimization:

$$\begin{aligned} \min_{w} \quad & \|w - \tilde{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i \\ \text{sb.t.} \quad & y_i \langle w, x_i \rangle \geq 1 - \xi_i, \quad \xi_i \geq 0 \text{ for all } 1 \leq i \leq n. \end{aligned} \tag{3.18}$$

By plugging in all these definitions in Theorem 6 and using a collection of standard tricks (described in Section 2.3) we obtain the following corollary:

Corollary 2. *For any $\delta > 0$ with probability at least $1 - \delta$ over the training sets $S_1, \ldots, S_T$ the following holds uniformly for all possible orders $\pi \in \mathcal{S}_T$ of T tasks:*

$$\frac{1}{2T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{(x,y)\sim D_t} [\![y \neq \operatorname{sign} \langle w_t, x \rangle]\!] \leq \frac{1}{T} \sum_{t=1}^{T} \left[\frac{1}{n} \sum_{i=1}^{n} \bar{\Phi} \left(\frac{y_i^{\pi(t)} \langle w_{\pi(t)}, x_i^{\pi(t)} \rangle}{||x_i^{\pi(t)}||} \right) + \right. \tag{3.19}$$

$$\left. \frac{||w_{\pi(t)} - w_{\pi(t-1)}||^2}{2\sqrt{n}} \right] + \frac{1}{8\sqrt{n}} - \frac{\log \delta}{T\sqrt{n}} + \frac{\log T}{\sqrt{n}}, \tag{3.20}$$

where $w_{\pi(0)} = \mathbf{0}$ is the zero vector and for every $t = 2, \ldots, T$ the weight vector $w_{\pi(t)}$ is obtained by solving (3.18) *using $S_{\pi(t)}$ and $w_{\pi(t-1)}$.*

Minimizing (3.20) requires searching over all possible permutations $\pi \in \mathcal{S}_T$. We propose to perform this search incrementally - for every t the task index $\pi(t)$ is determined

by optimizing the corresponding term in (3.20) with respect to the yet unsolved tasks $\tilde{T}$:

$$\pi(t) = \arg\min_{k \in \tilde{T}} \frac{1}{n} \sum_{i=1}^{n} \bar{\Phi}\left(\frac{y_i^k \langle v_k, x_i^k \rangle}{||x_i^k||}\right) + \frac{||v_k - w_{\pi(t-1)}||^2}{2\sqrt{n}}, \tag{3.21}$$

where every v_k is obtained by solving (3.18) using S_k and $w_{\pi(t-1)}$. Thus at every step the learner selects a task that is easy in the sense that it has low empirical error and is close to the last solved task in terms of ℓ_2 distance between the corresponding weight vectors. Therefore this procedure fits the intuition of starting with the simplest problem and then proceeding with the most similar ones. The method is summarized in Algorithm 1 and we refer to it as SeqMT. The computational complexity of SeqMT is quadratic in the number of tasks T, because at every step it trains an ASVM for every yet unsolved task (steps 5-7 in Algorithm 1).

Algorithm 1 Sequential Learning of Multiple Tasks

1: **Input** $S_1, \ldots, S_T$ {training sets}
2: $\pi(0) \leftarrow 0$, $w_0 \leftarrow \mathbf{0}$
3: $\tilde{T} \leftarrow \{1, 2, \ldots, T\}$ {indices of yet unused tasks}
4: **for** $t = 1$ to T **do**
5: **for all** $k \in \tilde{T}$ **do**
6: $v_k \leftarrow$ solution of (3.18) using S_k, $w_{\pi(t-1)}$
7: **end for**
8: $\pi(i) \leftarrow$ minimizer of (3.21) w.r.t. k
9: $w_{\pi(t)} \leftarrow v_k$ where $k = \pi(t)$
10: $\tilde{T} \leftarrow \tilde{T} \setminus \{\pi(t)\}$
11: **end for**
12: **Return** $w_1, \ldots, w_T$ and $\pi(1), \ldots, \pi(T)$

In order to verify that SeqMT can be used to find a favourable order of tasks we perform an experimental evaluation on the *Animals with Attributes (AwA)*[1] [49] and *Shoes*[2] [19] augmented with attributes[3] [46] datasets.

In the first experiment we focus on eight classes from the *AwA* dataset: *chimpanzee, giant panda, leopard, persian cat, hippopotamus, raccoon, rat, seal*, for which there is additional human annotation available [87]. For each class this annotation consists of ranking scores of its images, whether an object is *easy* or *hard* to recognize. We split the data in each class into five equal parts with respect to this ranking. Each part has

[1] http://attributes.kyb.tuebingen.mpg.de/
[2] http://tamaraberg.com/attributesDataset/index.html
[3] http://vision.cs.utexas.edu/whittlesearch/

on average 120 samples except the class *rat*, for which there are only approximately 60 samples per part. We create five tasks per class, where every problem is a binary classification of one of the parts against the remaining seven classes. For each task we randomly select 21 vs 21 training images and 77 vs 77 test images (35 vs 35 in case of class *rat*) with equal amount of samples from each of the classes acting as negative examples. We also make sure that the data between different tasks does not overlap. As our feature representation, we use 2000-dimensional bag-of-words histograms obtained from SURF descriptors [15], which we ℓ_2-normalize and augment with a unit element to act as a bias term. All the methods considered in the evaluation have one free parameter that we choose from 8 values $\{10^{-2}, 10^{-1} \ldots, 10^{5}\}$ using 5×5 fold cross-validation.

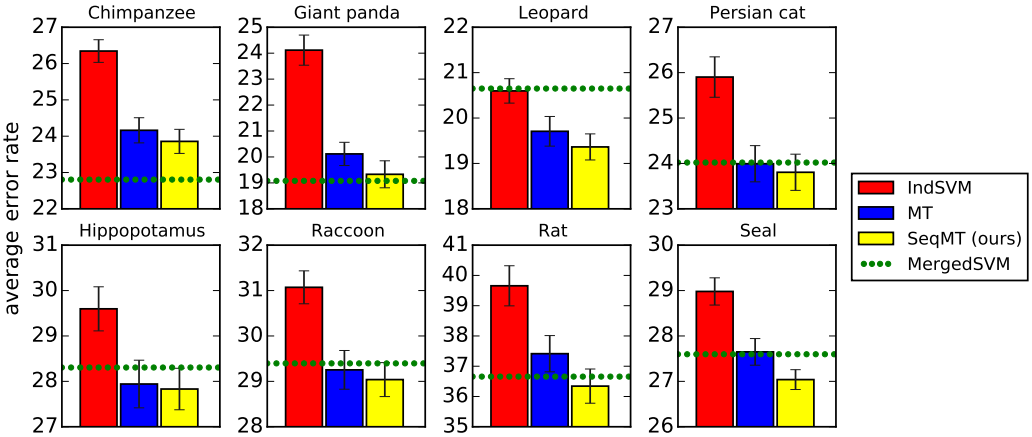


Figure 3.2: AwA dataset: comparison of SeqMT method with (3.10) (MT), the single-task (IndSVM) and MergedSVM baselines. The height of the bar corresponds to the average error rate (with standard error of the mean) over 5 tasks across 20 repeats.

In order to demonstrate potential advantages of the sequential approach to multi-task learning we compare SeqMT to the ordinary multi-task method (3.10), which we refer to as MT. As can be seen from Figure 3.2, the proposed approach outperforms MT method in all 8 cases. This supports the intuition that, if not all tasks are equally related, learning them sequentially can be more effective than jointly. The rather poor performance of a linear SVM baseline that solves each task independently (IndSVM) verifies that the parameter transfer approach is relevant in the considered setting. As a reference, we also trained a linear SVM that merges the data from all tasks and outputs one linear predictor for all tasks (MergedSVM). Its performance is rather unstable: in the cases of *chimpanzee* and *giant panda* MergedSVM outperforms SeqMT and MT methods, which indicates that the corresponding tasks are so similar that a single hyperplane

can explain most of them. In this case, MergedSVM benefits from the amount of data that is available to find this hyperplane. However, when the tasks are different enough, MergedSVM loses to SeqMT and MT models. In particular, in the case of *leopard* MergedSVM does not improve even over independent training.

In order to examine the effect that the task order has on the overall accuracy, we compare SeqMT to other methods that solve tasks sequentially using an Adaptive SVM (3.18) and differ only by how the tasks are ordered. First, as can be seen from Figure 3.3, SeqMT outperforms the Random baseline that processes tasks in a random order. We also evaluate the Semantic baseline that orders tasks from easiest to hardest according to human annotation as if it was given to the learner. In 6 out of 8 classes, SeqMT is better or on par with this baseline. Interestingly, in the case of *seal* and *hippopotamus* Semantic is on par or even worse than Random baseline. This indicates that the human intuition of what is a favorable order of tasks does not always coincide with what is beneficial for a machine learning system. In addition, we also evaluate a baseline inspired by the diversity heuristic from [80]. It defines the next task to be solved by maximizing (3.21) instead of minimizing it. We refer to it as Diversity. However this max heuristic is not effective in the considered setting. Lastly, since there are only five tasks in each experiment, we could also evaluate all possible deterministic orders of tasks, which results in 120 baselines. We visualize their performance using a violin plot [37], where the width of a horizontal slice of the shaded area reflects how many different orders achieve this error rate (performance stated on the vertical axis). In general, SeqMT is comparable to the best possible fixed orders. Interestingly, it outperforms them in two cases of *rat* and *seal*, which is possible because SeqMT may vary the order of tasks between different repeats, while every baseline corresponding to one of 120 task orders solves tasks in the same order in all 20 repeats. Thus, learning an adaptive order of tasks based on the training data may be advantageous to solving them in any fixed order, even the best one.

In the second experiment we focus on 10 classes of shoe models from the *Shoes* dataset [19]: *athletic, boots, clogs, flats, heels, pumps, rain boots, sneakers, stiletto, wedding shoes*, which are associated with 10 attributes [46]: *pointy at the front, open, bright in color, covered with ornaments, shiny, high at the heel, long on the leg, formal, sporty, feminine*. Attributes are provided per class by scores ranging from 1, denoting

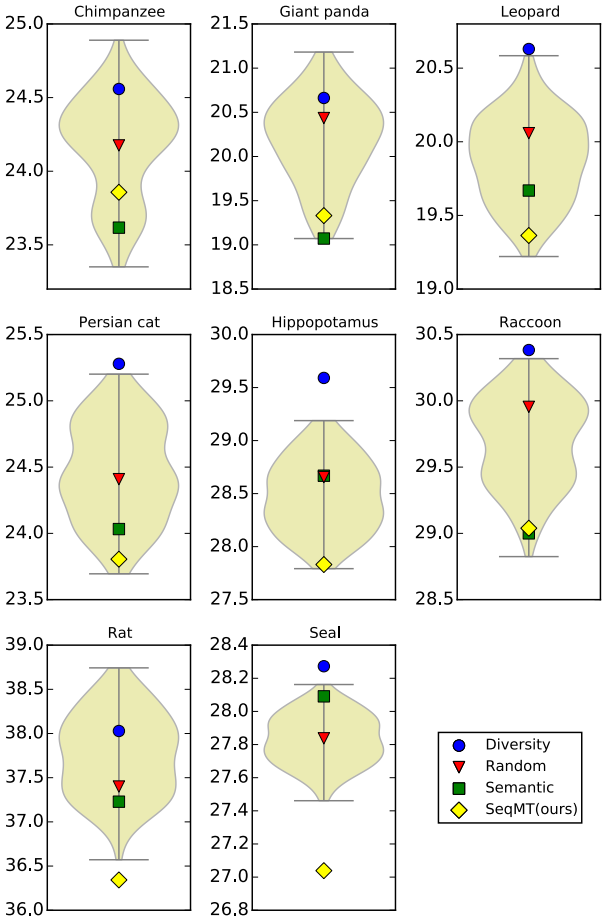


Figure 3.3: Different task order strategies in the experiment with *AwA* dataset. Averaged error rate performance (averaged over 20 repeats) is shown on the vertical axis.

class that "has it the least", to 10, denoting class that “has it the most”. Using this information for each attribute we form a binary classification task, using samples from classes with ranks 10 and 9 as positives and samples from classes with ranks 1 and 2 as negatives. As a result we obtain 10 tasks. For each of them we use 50 positive and 50 negative samples for training and 300 positive and 300 negative images for testing, which are randomly sampled from each class in equal amount. As features, we use 960 dimensional GIST descriptor concatenated with ℓ_1-normalized 30 dimensional color descriptor and augment them with a unit element as bias term.

We start with evaluating the same baselines as in the experiment on the *AwA* dataset: SepMT, MT, IndSVM, MergedSVM, Diversity and Random. As can be seen from Table 3.1, in contrast to the first experiment, none of the methods involving parameter transfer between the tasks show a significant improvement over the independent SVM approach. This might be because in this setting there are some tasks that are clearly

Methods	Average error
IndSVM	10.34 ± 0.13
MergedSVM	29.67 ± 0.10
MT	10.37 ± 0.13
Diversity	12.66 ± 0.17
Random	12.14 ± 0.20
SeqMT	10.96 ± 0.12

Table 3.1: Experiment on Shoes dataset. We report average error rate performance over 10 tasks across 20 repeats with standard error of the mean.

related such as *high heel* and *shiny*, and some tasks that are not, such as *high heel* and *sporty*. And at the same time all the considered methods do not allow for any groups of tasks being not related to each other.

The sequential transfer approach does not necessary need all the tasks to be related to each other. However it relies on the idea that the tasks can be ordered in such a way that each of them is related to the previous one. In practical applications and as was observed in the experiment on *Shoes* dataset this might not be the case since there might be some outlier tasks that are not similar to any other task, or there might be some groups in the underlying task structure such that only tasks within a group are related but there is no connection between the groups. In such cases forcing transfer between unrelated tasks may lead to a decrease in the performance. This problem can be avoided by allowing the learner to form multiple subsequences of tasks without forcing information transfer between them. Alternatively, one could think about a sequence of all tasks, but the learner is allowed to not transfer information between some of the subsequent tasks and use the original prior P instead, i.e. start a new subsequence. To describe this setting we introduce a set of flags $b_t \in \{0, 1\}$ for $t = 2, ..., T$, where $b_t = 1$ means that to solve the task $\pi(t)$ the learner uses the information provided by the previous tasks in the current subsequence, while $b_t = 0$ denotes that there is no transfer and P is used as a prior. As a result, we can prove the following modification of Theorem 6:

Theorem 7 ([74])**.** *For any deterministic transfer algorithm $\mathcal{TA}$, any deterministic learning algorithm $\mathcal{A}$, any prior P and any $\delta > 0$, the following holds with probability at least $1 - \delta$ over sampling the training sets $S_1, ..., S_T$ of size n each uniformly for all orders π in*

the symmetric group $\mathcal{S}_T$ and any set of flags $b_2, ..., b_T \in \{0, 1\}$:

$$\mathrm{er}_{\mathbf{D}}(\mathbf{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{Q}) + \frac{1}{T\sqrt{n}} \sum_{t=1}^{T} \mathrm{KL}(Q_{\pi(t)} \| P_{\pi(t)}) + + \frac{1 + 8 \log 2T}{8\sqrt{n}} + \frac{\log 1/\delta}{T\sqrt{n}}, \tag{3.22}$$

where:

$$\mathbf{Q} = (Q_1, \ldots, Q_T) \tag{3.23}$$

$$Q_{\pi(t)} = \mathcal{A}(P_{\pi(t)}, S_{\pi(t)}), \tag{3.24}$$

$$P_{\pi(t)} = \begin{cases} P \ \textit{for} \ t = 1 \ \textit{or} \ b_t = 0 \\ \mathcal{TA}(S_{\pi(i)}, \ldots, S_{\pi(t-1)}) \ \textit{for} \ b_t = 1 \ \textit{and} \ i = \max\{j : j < t \wedge b_j = 0\} \end{cases} \tag{3.25}$$

Analogously to Corollary 2, one obtains an instance of Theorem 7 for the case of linear predictors:

Corollary 3. *For any $\delta > 0$, the following holds with probability at least $1 - \delta$ over sampling the training sets $S_1, ..., S_T$ of size n each uniformly for all orders π of T tasks and any set of flags $\{b_2, ..., b_T\} \in \{0, 1\}^{T-1}$:*

$$\frac{1}{2T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{(x,y)\sim D_t} [\![y \neq \mathrm{sign}\langle w_t, x\rangle]\!] \leq \frac{1}{T} \sum_{t=1}^{T} \left[\frac{1}{n} \sum_{i=1}^{n} \bar{\Phi} \left(\frac{y_i^{\pi(t)} \langle w_{\pi(t)}, x_i^{\pi(t)} \rangle}{||x_i^{\pi(t)}||} \right) + \frac{||w_{\pi(t)} - b_t w_{\pi(t-1)}||^2}{2\sqrt{n}} \right] + \frac{1}{8\sqrt{n}} - \frac{\log \delta}{T\sqrt{n}} + \frac{\log 2T}{\sqrt{n}}, \tag{3.26}$$

where $w_{\pi(0)} = \mathbf{0}$ and for every $t = 2, \ldots, T$ $w_{\pi(t)}$ is obtained by solving (3.18) *using $S_{\pi(t)}$ and $b_t w_{\pi(t-1)}$.*

The corresponding algorithm that incrementally minimizes the right hand side of (3.26) is summarized in Algorithm 2 and we refer to it as MultiSeqMT. Like SeqMT, its multiple subsequences version, MultiSeqMT, chooses which task to solve next iteratively. However, at every step it has a possibility to continue any of the existing subsequences or even start a new one. Specifically, at step (6) of Algorithm 1 the learner for every yet unsolved task learns its weight vector w_t using the Adaptive SVM algorithm (3.18) not only with the weight $w_{\pi(t-1)}$ of the last solved task, but also with the weight vectors of the last tasks of all the existing subsequences and with zero vector.

Methods	Average error
IndSVM	10.34 ± 0.13
MergedSVM	29.67 ± 0.10
MT	10.37 ± 0.13
Diversity	12.66 ± 0.17
Random	12.14 ± 0.20
SeqMT (ours)	10.96 ± 0.12
MultiSeqMT (ours)	$\mathbf{9.95 \pm 0.12}$
RandomMultiSeq	10.89 ± 0.14

Table 3.2: Extended experiment on Shoes dataset. The numbers correspond to average error rate performance over 10 tasks across 20 repeats with standard error of the mean.

Therefore at every iteration (3.20) is used not only to define the next task to solve, but also to decide which subsequence to continue. Note that the possibility to generate multiple subsequences of tasks results in higher complexity of running MultiSeqMT as it may require to solve (3.18) up to T^3 times.

Algorithm 2 MultiSeqMT: Sequential Learning with Multiple Subsequences

Input $S_1, \dots, S_T$ {training sets}
$\tilde{T} \leftarrow \{1, 2, \dots, T\}$ {indices of yet unused tasks}
$P \leftarrow \{\mathbf{0}\}$ {ws of the last tasks in the existing subsequences}
for $t = 1$ to T **do**
  for all $\tilde{w} \in P$ **do**
    $k(\tilde{w}) \leftarrow$ steps 5-7 of Algorithm 1 with $\tilde{w}$ instead of $w_{\pi(t-1)}$
  end for
  $w^* \leftarrow$ minimizer of (3.20) w.r.t. $\tilde{w}$ with substituting $w_{\pi(t-1)}$ by $\tilde{w}$ and k by $k(\tilde{w})$
  $w_{k(w^*)} \leftarrow$ solution of (3.18) using $S_{k(w^*)}$ and w^*
  $\tilde{T} \leftarrow \tilde{T} \setminus \{k(w^*)\}$
  $P \leftarrow P \cup \{w_{k(w^*)}\}$
  if $w^* \neq \mathbf{0}$ **then**
    $P \leftarrow P \setminus \{w^*\}$
  end if
end for
Return $w_1, \dots, w_T$

In order to demonstrate potential benefits of the ability to divide tasks into subsequences, provided by MultiSeqMT, we evaluate it on the described before experiment with *Shoes* dataset, where all other parameter transfer approaches failed to outperform the independent SVM method. Additionally we include a baseline RandomMultiSeq that learns attributes in a random order with an option to randomly start a new subsequence.

As can be seen from Table 3.2, MultiSeqMT outperforms all other baselines, while SeqMT and MT are affected by forcing transfer between unrelated tasks and match the

performance of IndSVM. This confirms that learning multiple subsequences is advantageous, when not all given tasks are equally related. Moreover, learning multiple random subsequences as RandomMultiSeq does is better than learning a single sequence of all tasks, as SeqMT, Random and Diversity baselines do.

3.3 Active task selection

All multi-task methods mentioned before need at least some training data for all tasks of interest and focus on reducing the sample complexity per task. However, in some situations, the fixed costs of obtaining annotated examples can be high, while variable costs per label are reasonable. In such scenarios collecting even a few annotated examples for every task of interest might be expensive and it would be preferable to obtain possibly larger amounts of training data, but from fewer tasks. For example, for building a personalized speech recognition system it would be easier to collect a reasonable amount of data from a few users, rather than a few examples from every potential user of the system.

The active task selection approach is an alternative to more traditional multi-task methods that does not need access to annotated training data for all tasks of interest. In contrast, initially every task is represented only by a set of unlabeled examples. Based on this information, the learner is allowed to select a subset of tasks for which to request labels. After obtaining those, the learner needs to provide solutions for all tasks, selected as well as not selected. Similarly to active learning the hope is that selecting objects to be labeled in an intelligent, data-dependent way might be more beneficial than choosing them at random. However, in contrast to traditional active learning where the learner has to provide only one predictor for all examples, in active task selection he has to identify individual predictors for every task of interest, including those for which there are no labeled examples available. Such unlabeled tasks can be solved only by transferring information from the selected labeled tasks. This is a type of situation considered in unsupervised domain adaptation and the quality of the resulting solutions will depend on the choice of transfer algorithm. Moreover, the transfer method should also influence the choice of the labeled tasks, because for different approaches different subsets of tasks might be most informative. In this section we will describe how to select

the labeled tasks in a principled way for two domain adaptation methods.

We focus on binary classification with $0/1$ loss, i.e. $\mathcal{Y} = \{-1, 1\}$ and $\ell(y, y') = [\![y \neq y']\!]$. Moreover, we consider only deterministic labeling functions. Thus we assume that there are T tasks $\langle D_1, f_1\rangle, \ldots, \langle D_T, f_T\rangle$, where every task t is described by a marginal distribution D_t over the input space $\mathcal{X}$ and a labeling function $f_t : \mathcal{X} \to \mathcal{Y}$. Initially the learner is given a collection of T unlabeled training sets $S_1^u, \ldots, S_T^u$, where every $S_t^u = \{x_1^t, \ldots, x_n^t\}$ consists of n i.i.d. samples from the corresponding marginal distribution D_t. Based on this information the learner is allowed to select a subset $\{t_1, \ldots, t_k\}$ of k tasks. For each selected task t_i the learner obtains labels for a random subset $S_{t_i}^l \subset S_{t_i}^u$ of m points (we assume that these examples are sampled from the unlabeled training set without replacement).

Probably the simplest unsupervised domain adaptation method is to train a classifier on the labeled data from one task and use it without any changes on the task of interest. The expected performance of this hypothesis on the target task depends on how similar the tasks are in terms of the discrepancy [43, 17] between their marginal distributions:

Definition 6 (Definition 4 in [54])**.** The discrepancy between distributions D_1 and D_2 over $\mathcal{X}$ with respect to a hypothesis set $\mathcal{H}$ is defined as:

$$\operatorname{disc}(D_1, D_2) = \max_{h, h' \in \mathcal{H}} \left| \operatorname{er}_{D_1}(h, h') - \operatorname{er}_{D_2}(h, h') \right|, \tag{3.27}$$

where $\operatorname{er}_{D_i}(h, h') = \mathbb{E}_{x \sim D_i}\, \ell(h(x), h'(x))$.

The corresponding guarantees are provided by the following result:

Proposition 1 (Theorem 2 in [16])**.** *For any two tasks $\langle D_1, f_1\rangle$ and $\langle D_2, f_2\rangle$ and any hypothesis $h \in \mathcal{H}$ the following inequality holds:*

$$\operatorname{er}_2(h) \leq \operatorname{er}_1(h) + \operatorname{disc}(D_1, D_2) + \lambda_{12},$$

where $\lambda_{12} = \min_{h \in \mathcal{H}}(\operatorname{er}_1(h) + \operatorname{er}_2(h))$.

One of the important advantages of discrepancy as a measure of the difference between distributions is that it can be estimated based on the unlabeled samples:

Proposition 2 (Lemma 1 in [16]). *Let d be the VC dimension of the hypothesis set $\mathcal{H}$ and S_1, S_2 be two i.i.d. samples of size n from D_1 and D_2 respectively. Then for any $\delta > 0$ with probability at least $1 - \delta$:*

$$\operatorname{disc}(D_1, D_2) \leq \operatorname{disc}(S_1, S_2) + 2\sqrt{\frac{2d\log(2n) + \log(2/\delta)}{n}},$$

where

$$\operatorname{disc}(S_1, S_2) = \max_{h,h' \in \mathcal{H}} |\widehat{\operatorname{er}}_{S_1}(h, h') - \widehat{\operatorname{er}}_{S_2}(h, h')|$$

is the empirical discrepancy between the samples and

$$\widehat{\operatorname{er}}_{S_i}(h, h') = \frac{1}{n} \sum_{x \in S_i} \ell(h(x), h'(x)).$$

Assuming that this simple method is the transfer approach selected by the learner, active task selection approach reduces to choosing k tasks and assigning each of the remaining tasks to one of them based on the unlabeled data. We encode such an assignment by a vector $C = (c_1, \ldots, c_T)$ that has at most k different components corresponding to the selected tasks and c_t specifies which of them is used as a source of information for the t-th task. Thus the only remaining question is how to find such an assignment wisely based only on the unlabeled data. The following theorem quantifies the effect that the selection of the labeled tasks and the assignment of the unlabeled tasks to them have on the overall multi-task generalization error:

Theorem 8. *Let d be the VC dimension of the hypothesis set $\mathcal{H}$, k be the maximum number of tasks for which the learner may ask for labels, $S_1^u, \ldots, S_T^u$ be T random sets of size n each, where $S_t \overset{i.i.d.}{\sim} D_t$, and $S_1^l, \ldots, S_T^l$ be their random subsets of size m each, for which labels can be provided upon learner's request. Then, for any $\delta > 0$, provided that the choice of labeled tasks $I = \{t_1, \ldots, t_k\}$ and assignment $C = (c_1, \ldots, c_T)$ are fully determined by the unlabeled data, the following inequality holds with probability at least $1 - \delta$ uniformly for all possible choices of the assignment C and the corresponding*

hypotheses:

$$\frac{1}{T}\sum_{t=1}^{T}\mathrm{er}_t(h_{c_t}) \leq \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{er}}_{c_t}(h_{c_t}) + \frac{1}{T}\sum_{t=1}^{T}\mathrm{disc}(S_t, S_{c_t}) + \frac{1}{T}\sum_{t=1}^{T}\lambda_{tc_t} \tag{3.28}$$

$$+\sqrt{\frac{2d\log(em/d)}{m}} + \sqrt{\frac{\log(4/\delta)}{2m}} + \sqrt{\frac{2d\log(en/d)}{n}} + \sqrt{\frac{\log(T)+\log(4/\delta)}{2n}}$$

$$+\frac{2(T-k)}{T}\sqrt{\frac{2d\log(2n)+2\log(T)+\log(4/\delta)}{n}}. \tag{3.29}$$

where $\widehat{\mathrm{er}}_t(h) = \frac{1}{m}\sum_{x\in S_t^l}\ell(h(x), f_t(x))$.

The above theorem provides an upper bound on the average expected error on all tasks by the sum of three complexity terms and three task-dependent terms: training errors on the labeled tasks, average distances to the prototype in terms of the empirical discrepancies and an average of λ-s. The first complexity term vanishes as the number of unlabeled examples per task (n) tends to infinity, indicating that in this case the discrepancies between the tasks can be estimated precisely. If the number of labeled examples for each of the selected tasks (m) tends to infinity, the remaining complexity terms converge to 0 as $1/\sqrt{m}$, showing that in this case the learner has full knowledge about the labeled tasks. This rate of convergence is the best we can expect for the considered transfer method, because there is no sharing of information between the labeled tasks.

The only component of the right hand side of (3.29) that can not be estimated from the data that is available to the learner is the average of the λ-s with respect to the assignment C. While discrepancy captures the similarity between the marginal distributions, λ in addition embodies the similarity between the labeling functions. And though Theorem 8 holds without any assumptions on the task relatedness, its guarantees are most relevant when this term (as well as others) is small. While it is unreasonable to expect λ_{ij} to be small for every pair of tasks i and j, the right hand side of (3.29) is small (and thus informative) only if the average discrepancy between every unlabeled task and the labeled task it is assigned to is small:

$$\frac{1}{T}\sum_{t=1}^{T}\mathrm{disc}(S_t, S_{c_t}). \tag{3.30}$$

Thus requirement on the λ-term to be small can be reformulated as some kind of a

smoothness assumption: if two tasks have similar marginal distributions, i.e. discrepancy between them is small, they likely have similar labeling functions.

Minimizing (3.29) (taking into account that the choice of labeled tasks has to be done based only on the unlabeled data) results in the following strategy for the active task selection with single-source transfer (ATS-SS):

1. estimate pairwise discrepancies between the tasks based on the unlabeled data
2. minimize (3.30), i.e. cluster the tasks using the k-medoids method based on the obtained empirical discrepancies
3. train classifiers for the cluster centers and transfer them to the other tasks in the corresponding clusters.

Note that the inequality (3.29) holds uniformly with respect to assignment C and the corresponding hypotheses, thus it also holds for the output of ATS-SS.

Theorem 8 and the resulting approach ATS-SS are based on the assumption that for solving every task the learner uses only one of the chosen labeled tasks and there is no information transfer between the labeled tasks. The simplicity of this approach makes it intuitive and allows for simple analysis and implementation. However, it might be suboptimal: potentially all k labeled tasks could be used to obtain predictors for the remaining unlabeled tasks and information could also be shared between them. In order to exploit this possibility, we consider a domain adaptation method [16] that minimizes a convex combination of training errors on all given source domains, i.e. labeled tasks.

Given a set of tasks $I = \{t_1, \ldots, t_k\} \subset \{1, \ldots, T\}$ define:

$$\Lambda^I = \left\{ \alpha \in [0,1]^T : \sum_{i=1}^{T} \alpha_i = 1;\ \operatorname{supp} \alpha \subseteq I \right\} \tag{3.31}$$

for $\operatorname{supp} \alpha = \{i \in \{1, \ldots, T\} : \alpha_i \neq 0\}$. For a weight vector $\alpha \in \Lambda^I$, an α-weighted empirical error of a hypothesis $h \in \mathcal{H}$ is defined as follows:

$$\widehat{\mathrm{er}}_\alpha(h) = \sum_{i \in I} \alpha_i \widehat{\mathrm{er}}_i(h). \tag{3.32}$$

Now we consider the scenario when in order to obtain a predictor for every task t, labeled as well as unlabeled, the learner minimizes $\widehat{\mathrm{er}}_{\alpha^t}(h)$ for some parameter vector $\alpha^t \in \Lambda^I$,

where I is the set of selected labeled tasks. This is a generalization of the previously considered transfer from a single labeled task, because it reduces to that method if every weight vector α^t has only one non-zero component. However, real-valued weights can potentially improve the performance and the following theorem provides a guide on how to choose which tasks to label for this adaptation method:

Theorem 9. *Let d be the VC dimension of the hypothesis set $\mathcal{H}$, k be the maximum number of tasks for which the learner may ask for labels, $S_1^u, \ldots, S_T^u$ be T sets of size n each, where $S_i^u \overset{i.i.d.}{\sim} D_i$, and $S_1^l, \ldots, S_T^l$ be their random subsets of size m each for which labels would be provided upon learner's request. Then for any $\delta > 0$, provided that the choice of labeled tasks $I = \{i_1, \ldots, i_k\}$ and the weights $\alpha^1, \ldots, \alpha^T \in \Lambda^I$ are fully determined by the unlabeled data only, the following inequality holds with probability at least $1 - \delta$ over $S_1^u, \ldots, S_T^u$ and $S_1^l, \ldots, S_T^l$ for all possible choices of I, $\alpha^1, \ldots, \alpha^T \in \Lambda^I$ and $h_1, \ldots, h_T \in \mathcal{H}$:*

$$\frac{1}{T}\sum_{t=1}^{T} \mathrm{er}_t(h_t) \leq \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathrm{er}}_{\alpha^t}(h_t) + \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in I} \alpha_i^t \,\mathrm{disc}(S_t, S_i) \tag{3.33}$$

$$+ \frac{A}{T}\|\alpha\|_{2,1} + \frac{B}{T}\|\alpha\|_{1,2} + C + D + \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in I} \alpha_i^t \lambda_{ti}, \tag{3.34}$$

where:

$$\|\alpha\|_{2,1} = \sum_{t=1}^{T} \sqrt{\sum_{i\in I} (\alpha_i^t)^2}, \quad \|\alpha\|_{1,2} = \sqrt{\sum_{i\in I} \left(\sum_{t=1}^{T} \alpha_i^t\right)^2},$$

$$A = \sqrt{\frac{2d\log(ekm/d)}{m}}, \quad B = \sqrt{\frac{\log(4/\delta)}{2m}},$$

$$C = \sqrt{\frac{8(\log T + d\log(enT/d))}{n}} + \sqrt{\frac{2}{n}\log\frac{4}{\delta}},$$

$$D = 2\sqrt{\frac{2d\log(2n) + 2\log(T) + \log(4/\delta)}{n}}.$$

First, note that in the worst case, when there exists $i \in I$ such that $\alpha_i^t = 1$ for every $t \in \{1, \ldots, T\}$, $\|\alpha\|_{2,1} = \|\alpha\|_{1,2} = T$ and, thus, the term $\frac{A}{T}\|\alpha\|_{2,1} + \frac{B}{T}\|\alpha\|_{1,2}$ behaves as

$O(\sqrt{d\log(km)/m})$. In the opposite extreme, if every α^t weights all the labeled tasks equally, i.e. $\alpha_i^t = 1/k$ for all $t \in \{1,\dots,T\}$ and $i \in I$, $\|\alpha\|_{2,1} = \|\alpha\|_{1,2} = \frac{T}{\sqrt{k}}$. Therefore the convergence of the corresponding term improves from $O(\sqrt{d\log(km)/m})$ to $O(\sqrt{d\log(km)/km})$, which is the best we can expect from having km labeled examples. Thus, this term in (3.35) captures the intuition that multi-source approach may improve the performance: it encourages the learner to use data from multiple labeled tasks for adaptation and to select the tasks that all would be equally useful, thus preventing labeling tasks that would be useful for only a few others.

At the same time the complexity terms C and D behave as $O(\sqrt{d\log(nT)/n})$. In order for these terms to be balanced with $\frac{A}{T}\|\alpha\|_{2,1} + \frac{B}{T}\|\alpha\|_{1,2}$, i.e. for the uncertainty coming from the estimation of discrepancy to not dominate the uncertainty from the estimation of the α-weighted risks, the number of unlabeled examples per task n should be significantly (for $k \ll T$) larger than m. However, under the common assumption that obtaining enough unlabeled examples is significantly cheaper than annotated ones, this is not a strong limitation.

As in the case of single-source transfer, we can use the right hand side of Theorem 9 to guide the choice of the labeled tasks and the weights $\alpha^1,\dots,\alpha^T$. In particular, its part that can be evaluated based only on the unlabeled data and depends on the choice of I and α-s is:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i\in I}\alpha_i^t \operatorname{disc}(S_t, S_i) + \frac{A}{T}\|\alpha\|_{2,1} + \frac{B}{T}\|\alpha\|_{1,2}. \tag{3.35}$$

Thus we obtain the following analog of ATS-SS for the case of multi-source transfer (ATM-MS):

1. estimate pairwise discrepancies between the tasks based on the unlabeled data
2. choose the labeled tasks I and the weights $\alpha^1,\dots,\alpha^T$ by minimizing (3.35)
3. for every task t train a classifier by minimizing (3.32) using the obtained weight vector α^t.

Note that the above method satisfies the conditions of Theorem 9, thus its guarantees also hold for the resulting solution.

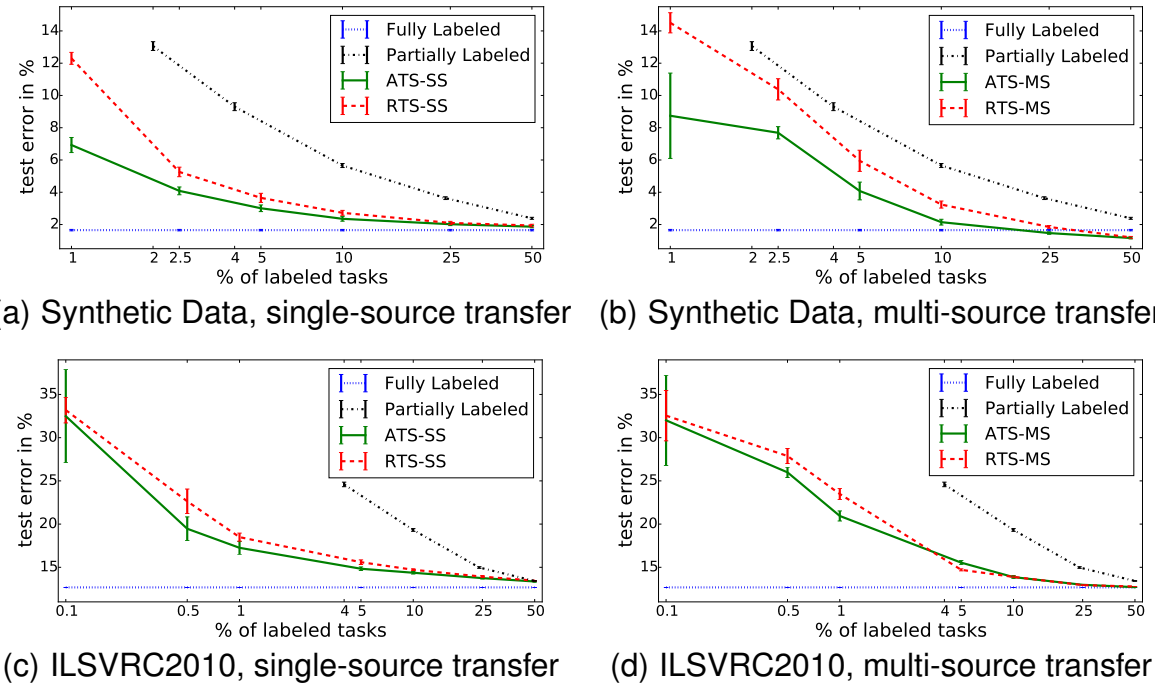
Both ATS-SS and ATS-MS come with theoretical guarantees. However, implementing them exactly is computationally hard, because it requires performing ERM and solving

k-medoids problem, both of which are in general NP-hard [41, 67]. Thus examine the performance of the variations of ATS-SS and ATS-MS in practice when the optimization problems are solved approximately by evaluating them on synthetic and real data. In both experiments we use linear predictors without a bias term.

Since there are no earlier methods for multi-task learning with unlabeled tasks, we compare both methods to the natural baseline commonly used to benchmark active learning methods: it selects the labeled tasks randomly and then applies the same adaptation method, i.e. each unlabeled task is solved using the predictor from the closest labeled task (RTS-SS), or by training on a task-specific weighted combination of labeled tasks (RTS-MS) with only the weights obtained by minimizing (3.35). We also evaluate independent ridge regressions that have access to labels for all tasks (denoted by *Fully Labeled*). However, this baseline has access to many more annotated examples in total than the active and random task selection methods. In order to quantify this effect we evaluate a *Partially Labeled* baseline. This method also has access to labeled examples for all tasks, but the total number of annotated examples it sees is the same as for task selection approaches. In particular, when the number of labeled tasks is k, the number of labels per task the Partially Labeled baseline sees is mk/T. To avoid the need for heuristic choices, we report results for this baseline only for integer values of mk/T.

In order to estimate the empirical discrepancies between a pair of tasks (step 1 in ATS-SS and ATS-MS) we find a linear predictor that minimizes the squared loss for the binary classification problem of separating the two sets of instances, as in [16]. To minimize the k-medoid risk (step 2 in ATS-SS) we use local search [68]. For the corresponding minimization of (3.35) in ATS-MS we use the GraSP algorithm [9]. GraSP requires as a subroutine a method for optimizing the objective with respect to a given sparsity pattern, for which we use gradient descent. To obtain classifiers for the individual tasks in all scenarios we use least-squares ridge regression with regularization constant from the set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ found by 5×5-fold cross validation.

First, we generate synthetic data that well suits the active task selection paradigm. We construct $T = 1000$ binary classification tasks in $\mathbb{R}^2$, where every marginal distribution D_t is a unit-variance Gaussian with mean μ_t chosen uniformly at random from the set $[-5, 5] \times [-5, 5]$. The label $+1$ is assigned to all points that have angle between 0



(a) Synthetic Data, single-source transfer (b) Synthetic Data, multi-source transfer

(c) ILSVRC2010, single-source transfer (d) ILSVRC2010, multi-source transfer

Figure 3.4: Experimental results on synthetic and real data: average test error and standard deviation over 100 repeats (synthetic) or 20 repeats (real) for the proposed active task selection (ATS) and random task selection (RTS) as well as fully supervised and partially labeled baselines.

and π with μ_t (computed counter-clockwise), the other points are labeled -1. We use $n = 1000$ unlabeled and $m = 100$ labeled examples per task.

For the real-data experiment we use the train part of the ImageNet ILSVRC2010 dataset [79], which consists of approximately 1.2 million natural images from 1000 classes. We extract features using a deep convolutional neural network [88] that was pretrained on MIT Places. For computational simplicity we reduce their dimension to 5 using PCA and augment them with a constant feature, resulting in $d = 6$. We construct 999 balanced binary tasks of classifying the largest class, *Yorkshire terrier*, versus one of the remaining classes. We use $n = 400$ unlabeled samples per task and label a subset of $m = 100$ examples for each of the selected tasks.

As can be seen from Figure 3.4 in both single-source and multi-source adaptation cases choosing labeled tasks actively according to ATS-SS and ATS-MS is advantageous over a random choice, especially when the budget only allows for a small fraction of tasks to be labeled. The difference with the Partially Labeled baseline is even bigger, indicating that in this case, having more labels for fewer tasks rather than only a few labels for all tasks is beneficial not only in terms of annotation costs, but also in terms of

prediction accuracy. Note that if the tasks in question were completely unrelated this would likely not hold.

As the number of labeled tasks gets larger, e.g. half of all tasks, the performance of the active task selection learner becomes almost identical to the performance of the fully supervised method, even improving over it in the case of multi-source transfer on synthetic data. This confirms the intuition that in the case of many related tasks even a fraction of the tasks can contain enough information for solving all tasks.

3.4 Conclusion

In this chapter we have discussed various assumptions on task relatedness used in multi-task learning. In particular, we have shown that previous results in representation transfer that aim at learning a linear feature transformation can be extended to kernel learning. The obtained results, given by Theorem 5 show that access to data from several related learning tasks reduces the sample complexity per task for learning a kernel whenever the pseudodimension of the considered kernel family is finite. In the case of parameter transfer we have shown that processing tasks sequentially can be more effective than doing it jointly. Theorems 6 and 7 capture the effect of the task order on the average performance of the learner and can be used to derive algorithms capable to automatically determine a favorable order of tasks based only on the training data. We have illustrated this process in the case of linear predictors and shown the effectiveness of the resulting SeqMT and MultiSeqMT methods on two real-worlds image datasets. Finally, in Section 3.3 we have discussed the active task selection framework: a modification of the standard multi-task scenario when initially all tasks of interest are represented only by unlabeled data and the learner is able to select a subset of tasks for which to request labels. Theorems 8 and 9 provide analysis of this framework for two domain adaptation methods and can be used to make the choice of which tasks to label in a principled way. We have illustrated the performance of the resulting ATS-SS and ATS-MS methods on two datasets.

4 Lifelong Learning

In lifelong learning [91] the learner encounters a stream of tasks - $D_1, \ldots, D_T, \ldots$ For every task he observes a training set $S_t = \{(x_1^t, y_1^t), \ldots, (x_n^t, y_n^t)\}$ of annotated examples sampled i.i.d. from the corresponding distribution D_t. As in multi-task learning, we assume that all tasks share the same domain $\mathcal{X} \times \mathcal{Y}$ and the learner uses the same loss function ℓ to evaluate the performance of prediction methods. However, in contrast to the multi-task setting, at every time step T the goal of the learner is not to perform well on the observed tasks $D_1, \ldots, D_T$, but rather to extract some information from them that would be useful for solving new, yet unobserved tasks. Thus, for this goal to make sense one has to assume some relatedness between the observed tasks and the new ones.

The first formal model of the lifelong learning setting was proposed by Baxter [14], who introduced the notion of *task environment* - a set $\mathfrak{T}$ of all tasks that may need to be solved at some point and a probability distribution $\mathfrak{D}$ over it. Under Baxter's model the observed tasks are assumed to be sampled i.i.d. from some unknown task environment and the goal of the learner is to perform well in expectation over new tasks coming from the same environment. Thus, the task environment provides a way to formally define a process that generates the tasks. However, this is not enough. Imagine a situation where $\mathfrak{T}$ contains all possible tasks (or just sufficiently many) on a given domain and $\mathfrak{D}$ assigns to all of them the same probability. In such case the finitely many observed tasks very likely will not have anything in common and will not contain any useful information for the future. Thus, as in multi-task learning, one also has to assume some kind of functional relatedness between the tasks, which in the case of lifelong learning is formulated not on the level of the observed tasks, but on the level of the task

environment as a whole.

Conceptually almost all relatedness assumptions that are useful in multi-task learning, can also be exploited in lifelong learning by just casting the observed tasks at every time step T as a multi-task problem. Consider, for example, an assumption of a shared low-dimensional representation discussed in the multi-task scenario. Under this assumption the lifelong learner could just perform optimization (3.5) on the observed data, use the inferred feature representation for solving new tasks and repeat the process when new data arrives. The drawback of such a naive approach is that it requires re-training a model from scratch with every new task. This might not be satisfactory when dealing with a potentially long stream of tasks, where one would prefer to be able to efficiently update the model when new data arrives. Motivated by this observation, a modification of multi-task methods of representation transfer [5, 6] for lifelong learning was developed in [81, 82]. The main difference of the proposed there algorithm to its multi-task ancestor is that it allows for such fast, incremental updates.

From the theoretical perspective the difference between multi-task and lifelong learning is that in the latter case there is an additional source of uncertainty, because the learner does not know precisely what new tasks are going to be. In case of Baxter's assumption of a shared inductive bias [14] it results in searching for a hypothesis set $\mathcal{H} \in \mathbb{H}$ that minimizes the following, *lifelong* expected error:

$$\mathrm{er}_{\mathfrak{D}}(\mathcal{H}) = \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \mathrm{er}_D(\mathcal{H}) = \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \inf_{h\in\mathcal{H}} \mathop{\mathbb{E}}_{(x,y)\sim D} \ell(h(x), y) \tag{4.1}$$

instead of its multi-task version:

$$\mathrm{er}_{\mathbf{D}}(\mathcal{H}) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{er}_{D_t}(\mathcal{H}) = \frac{1}{T}\sum_{t=1}^{T} \inf_{h\in\mathcal{H}} \mathop{\mathbb{E}}_{(x,y)\sim D_t} \ell(h(x), y). \tag{4.2}$$

However, typically generalization bounds for lifelong expected error (4.1) are obtained by first relating it to the multi-task version (4.2) and then relating the latter one to its empirical counterpart. Thus, generalization bounds for multi-task learning can often be obtained as a subproduct of proving bounds for the lifelong learning scenario [57].

We take the same path to extend our result for multi-task multiple kernel learning to lifelong learning. In this case the quantity of interest - lifelong expected error - can be

written as:

$$\mathrm{er}_{\mathfrak{D}}(\mathcal{H}_K) = \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \inf_{h\in\mathcal{H}_K} \mathrm{er}_D(h) = \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \inf_{h\in\mathcal{H}_K} \mathop{\mathbb{E}}_{(x,y)\sim D} [\![yh(x) < 0]\!]. \tag{4.3}$$

The following extension of Theorem 5 provides a uniform bound on its deviation from the empirical counterpart given by:

$$\widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathcal{H}_K) = \frac{1}{T}\sum_{t=1}^{T} \inf_{h\in\mathcal{H}_K} \frac{1}{n}\sum_{i=1}^{n} [\![y_i^t h(x_i^t) < \gamma]\!]. \tag{4.4}$$

Theorem 10 (Theorem 5 in [70])**.** *Let $\langle \mathfrak{T}, \mathfrak{D}\rangle$ be a task environment defined on $\mathcal{X} \times \{-1, +1\}$ and $\mathcal{K}$ be any kernel family with finite pseudodimension $d_\phi(\mathcal{K})$, such that $K(x, x) \leq B^2$ for any $K \in \mathcal{K}$ and any $x \in \mathcal{X}$. Then, for any fixed $\gamma > 0$ and any $\epsilon > 0$, if $T > 8/\epsilon^2$ and $n > 8/\epsilon^2$, then:*

$$\Pr\left\{\forall K \in \mathcal{K}\ \ \mathrm{er}^{2\gamma}_{\mathfrak{D}}(\mathcal{H}_K) + \epsilon \geq \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathcal{H}_K) \geq \mathrm{er}_{\mathfrak{D}}(\mathcal{H}_K) - \epsilon\right\} \geq 1 - \delta, \tag{4.5}$$

where

$$\begin{aligned}
\mathrm{er}^{2\gamma}_{\mathfrak{D}}(\mathcal{H}_K) &= \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \inf_{h\in\mathcal{H}_K} \mathrm{er}^{2\gamma}_{D}(h) = \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \inf_{h\in\mathcal{H}_K} \mathop{\mathbb{E}}_{(x,y)\sim D} [\![yh(x) < 2\gamma]\!] \\
\delta &= 2^{T+2}\left(\frac{512eT^2n^3B^2}{\gamma^2 d_\phi}\right)^{d_\phi}\left(\frac{512nB^2}{\gamma^2}\right)^{\frac{1024B^2T}{\gamma^2}\log\left(\frac{e\gamma n}{16B}\right)}\exp\left(-\frac{Tn\epsilon^2}{32}\right) + \\
&\quad 4\left(32CT^5 d_\phi^5\left(\frac{64B}{\epsilon\gamma}\right)^{17}\right)^{d_\phi}\exp\left(-\frac{T\epsilon^2}{128}\right).
\end{aligned} \tag{4.6}$$

First note that, as in Theorem 5, the above result holds without any assumptions on the task environment $\langle \mathfrak{T}, \mathfrak{D}\rangle$. However, it has the most significant implications in the case when there exists some kernel $K \in \mathcal{K}$ that has low approximation error for all tasks in the environment. This is the assumption on task relatedness that is indirectly exploited by Theorem 10. In such case, the kernel that minimizes the average error over the set of observed tasks is expected to be useful in expectation for new tasks coming from the same environment.

The only difference between Theorem 10 and Theorem 5 is the second term in (4.6).

Technically, this term comes from bounding the difference between lifelong expected error $\mathrm{er}_{\mathfrak{D}}(\mathcal{H}_K)$ and its multi-task version:

$$\mathrm{er}_{\mathbf{D}}^{\gamma/2}(\mathcal{H}_K) = \frac{1}{T}\sum_{t=1}^{T} \inf_{h\in\mathcal{H}_K} \mathop{\mathbb{E}}_{(x,y)\sim D_t} [\![yh(x) < \gamma/2]\!]. \tag{4.7}$$

Indeed, the same arguments as for proving Theorem 5 can be used to bound the difference between $\mathrm{er}_{\mathbf{D}}^{\gamma/2}(\mathcal{H}_K)$ and its empirical counterpart $\mathrm{er}_{\mathbf{S}}^{\gamma}(\mathcal{H}_K)$. The only remaining step is to relate $\mathrm{er}_{\mathfrak{D}}(\mathcal{H}_K)$ to $\mathrm{er}_{\mathbf{D}}^{\gamma/2}(\mathcal{H}_K)$. Thus, this additional term in (4.6) captures the uncertainty coming from the fact that the lifelong learner does not know exactly what new tasks are going to be. In the limit of infinitely many observed tasks ($T \to \infty$) this term vanishes, indicating that by observing sufficiently many tasks the learner gets the full knowledge about the task environment. The first term in (4.6) behaves exactly the same as the one in Theorem 5: its part that depends on the pseudo-dimension d_ϕ vanishes as the number of observed tasks T grows and thus it converges to the complexity of learning one task as if the learner would know a good kernel in advance. The combination of two types of complexity terms - one for the task environment, i.e. the first expectation in the definition of the lifelong expected error, and one for the data distributions corresponding to the observed tasks - and vanishing of the overhead associated with inferring the commonality, i.e. kernel function in case of Theorem 10, are two distinct features of generalization bounds for lifelong learning that demonstrate complexity and potential benefits of this setting from the theoretical perspective.

4.1 PAC-Bayesian perspective

4.1.1 I.i.d. tasks

Baxter's idea of learning an inductive bias from multiple tasks can naturally be translated to PAC-Bayesian language. Though Theorem 4 holds regardless of the agreement between the prior distribution P and the underlying data distribution D, its implications are most significant when the prior is informative, i.e. is close to posteriors with low empirical error $\widehat{\mathrm{er}}_S(Q)$. In such cases the value of the right hand side of (2.19) can be made small, because there exist posteriors with low empirical error and small

KL-divergence from the prior.

While informative priors are most useful, in practice it might be hard to come up with one. Thus, it would be preferable to be able to infer an informative prior from the data. However, the only condition that the prior has to satisfy is to be independent from the training data. A way around this problem was proposed in [69] by splitting the data into two parts and using one for estimating a prior and another one for learning a predictor. However, this approach requires a significant amount of training data and can not be applied to lifelong learning, where the learner is interested in new, yet unobserved tasks. Alternatively, one could follow the logic of Baxter's inductive bias learning and try to infer the prior from multiple observed tasks. In this section we will discuss how to do this in a principled way. These results were published in the paper "A PAC-Bayesian Bound for Lifelong Learning" [71].

First, note that technically minimizing a PAC-Bayesian bound like (2.19) with respect to P is not a valid strategy for learning a prior. This is because, in contrast to being uniform in posterior Q, inequality (2.19) holds only for a fixed, data independent prior. Thus, if we wish to develop a PAC-Bayesian bound that can be used for learning priors from the data, its guarantees have to be uniform in P. In order to achieve this we treat the prior P itself as a random variable. We let $\mathcal{P}$ be an initial *hyperprior* distribution over the set of possible priors and reformulate the goal of inferring the prior from the observed tasks as adjusting $\mathcal{P}$ info a data-dependent *hyperposterior* distribution $\mathcal{Q}$ over the set of priors. We also assume that for every task t the learner uses a fixed deterministic learning algorithm that outputs a posterior $Q_t(S_t, P)$ based on a prior P and a training set S_t. Then the goal of the learner is to identify a hyperposterior $\mathcal{Q}$ that in expectation leads to a low expected error $\mathrm{er}_D(Q)$ on a new randomly sampled task D with a training set S, when the prior is sampled according to $\mathcal{Q}$:

$$\mathrm{er}_{\mathfrak{D}}(\mathcal{Q}) = \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \mathop{\mathbb{E}}_{S\sim D^n} \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \mathrm{er}_D(Q(S,P)). \tag{4.8}$$

Following the PAC-Bayesian path we require the hyperprior $\mathcal{P}$ to be independent from the data and obtain the following result that bounds the difference between $\mathrm{er}_{\mathfrak{D}}(\mathcal{Q})$ and

its empirical counterpart:

$$\widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q}) = \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \widehat{\mathrm{er}}_{S_t}(Q_t(S_t, P)), \tag{4.9}$$

uniformly for all hyperposteriors $\mathcal{Q}$:

Theorem 11 ([71])**.** *Let $\langle \mathfrak{T}, \mathfrak{D}\rangle$ be any task environment and $\mathcal{P}$ be any fixed hyperprior distribution. Then, for any $\delta > 0$ the following inequality holds with probability at least $1-\delta$ (over the training samples $\{S_1, \ldots, S_T\}$) for all hyperposterior distributions $\mathcal{Q}$:*

$$\begin{aligned} \mathrm{er}_{\mathfrak{D}}(\mathcal{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q}) + \left(\frac{1}{\sqrt{T}} + \frac{1}{T\sqrt{n}}\right) \mathrm{KL}(\mathcal{Q}\|\mathcal{P}) \\ + \frac{1}{T\sqrt{n}} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \mathrm{KL}(Q_t(S_t, P)\|P) + \frac{1}{\sqrt{n}}\left(\frac{1}{8} - \frac{1}{T}\log\frac{\delta}{2}\right). \end{aligned} \tag{4.10}$$

Like Theorem 10, Theorem 11 contains two types of complexity terms. The first one, $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ corresponds to the level of the task environment in general and vanishes only when the number of observed tasks T tends to infinity, i.e. when the learner has full information about the task environment $\langle \mathfrak{T}, \mathfrak{D}\rangle$. In contrast, in the second complexity term every $\mathrm{KL}(Q_t(S_t, P)\|P)$ belongs specifically to the t-th task. When T grows, this term converges to an average KL-divergence over tasks and may remain non-zero, indicating that full knowledge about the task environment is not sufficient to overcome the uncertainty within each task. When, in contrast, the number of samples per task n tends to infinity, the second complexity term converges to 0 as $1/\sqrt{n}$, while the first one does not, since there is still uncertainty on the task environment level.

As a PAC-Bayesian bound, the right hand side of (4.10) contains only computable quantities and can thus be seen as a quality measure of the hyperposterior $\mathcal{Q}$. By minimizing it one can obtain a hyperposterior that is well suited to the particular task environment and since (4.10) holds uniformly in $\mathcal{Q}$, the obtained distribution over the priors can be expected to work well on new tasks from the same environment. Now, we will illustrate this process for two information transfer strategies discussed in the multi-task chapter - parameter and representation transfer.

We focus on the case of linear predictors, i.e. $\mathcal{X} \subset \mathbb{R}^d$ and $h(x) = \langle w, x\rangle$ if $\mathcal{Y} = \mathbb{R}$

or $h(x) = \operatorname{sign}\langle w, x\rangle$ if $\mathcal{Y} = \{-1, 1\}$, where $w \in \mathbb{R}^d$ is a weight vector. We start with the parameter transfer approach that is based on the assumption that the weight vectors for different tasks are similar to each other [32]. In accordance with standard techniques, we let prior and posterior distributions be Gaussian with unit variance that differ only by the values of their means:

$$P = \mathcal{N}(w_P, I_d) \qquad \text{and} \qquad Q = \mathcal{N}(w_Q, I_d).$$

Thus priors are parametrised by w_P, which is first distributed according to the hyperprior distribution, $\mathcal{P} = \mathcal{N}(0, \sigma I_d)$ and later according to the hyperposterior $\mathcal{Q} = \mathcal{N}(w_\mathcal{Q}, I_d)$. To capture the relatedness assumption we use as a learning algorithm for every task the following modification of Adaptive SVM (3.18) with squared loss:

$$w_Q = \arg\min \left(\|w - w_P\|^2 + \frac{C}{n} \sum_{i=1}^{n} (y_i - \langle w, x_i\rangle)^2 \right), \tag{4.11}$$

which has a closed form solution

$$w_Q = \left(\frac{n}{C} I_d + XX^\top\right)^{-1} \left(\frac{n}{C} w_P + XY\right) = Aw_P + b, \tag{4.12}$$

where X is the matrix with columns $x_1, \ldots, x_n$, Y is a column of labels $(y_1, \ldots, y_n)^\top$, $A = \left(I_d + \frac{C}{n} XX^\top\right)^{-1}$ and $b = \frac{C}{n} AXY$.

Using these definitions we can compute the complexity terms in (4.10):

$$\begin{aligned} &\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) = \frac{\|w_\mathcal{Q}\|^2}{2\sigma} + \frac{d}{2}\Big(\log\sigma + \frac{1}{\sigma} - 1\Big) \quad \text{and} \\ &\mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \mathrm{KL}(Q_t(S_t, P)\|P) = \frac{1}{2}\Big(\|(A_t - I_d)w_\mathcal{Q} + b_t\|^2 + \operatorname{tr}(A_t - I_d)^2\Big). \end{aligned} \tag{4.13}$$

For the loss function ℓ we consider two cases - $0/1$ loss for binary classification and truncated squared loss $\ell(y, y') = \min\{(y - y')^2, 1\}$ for regression. In the first case the expected empirical error of the Gibbs classifier is given by the following expression [33, 50]

$$\widehat{\mathrm{er}}_\mathbf{S}(w_\mathcal{Q}) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} \bar{\Phi}\left(\frac{y_i^t (x_i^t)^\top (A_t w_\mathcal{Q} + b_t)}{\sqrt{(x_i^t)^\top (I_d + A_t A_t^\top) x_i^t}} \right). \tag{4.14}$$

Keeping in mind that for $0/1$-loss the Gibbs error is at most twice smaller than the

expected error of the classifier defined by $A_i w_{\mathcal{Q}} + b_i$ [63, 51], we obtain the following instantiation of inequality (4.10):

$$\forall w_{\mathcal{Q}} \quad \frac{1}{2} \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \mathop{\mathbb{E}}_{S\sim D^n} \mathop{\mathbb{E}}_{(x,y)\sim D} [y \neq \operatorname{sign}\langle A w_{\mathcal{Q}} + b, x\rangle] \leq \frac{\sqrt{Tn}+1}{2\sigma T\sqrt{n}} \|w_{\mathcal{Q}}\|^2 \tag{4.15}$$
$$+ \frac{1}{2T\sqrt{n}} \sum_{t=1}^{T} \|(A_t - I_d) w_{\mathcal{Q}} + b_t\|^2 + \frac{1}{T} \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} \bar{\Phi}\left(\frac{y_i^t (x_i^t)^\top (A_t w_{\mathcal{Q}} + b_t)}{\sqrt{(x_i^t)^\top (I_d + A_t A_t^\top) x_i^t}} \right) + const.$$

For regression tasks, since $\ell(y, y') = \min\{(y - y')^2, 1\} \leq (y - y')^2$, we can upper bound $\widehat{\mathrm{er}}(w_{\mathcal{Q}})$ by the empirical error of the Gibbs predictor with squared loss. This error differs from the error of the predictor that is defined by $A_t w_{\mathcal{Q}} + b_t$ only by an additive constant that does not depend on $w_{\mathcal{Q}}$. Moreover, similarly to $0/1$ loss, for truncated squared loss ℓ the error of Gibbs predictor is also at least one half of the expected error of the predictor defined by $A_t w_{\mathcal{Q}} + b_t$. Thus we obtain the following instantiation of (4.10) for the truncated squared loss:

$$\forall w_{\mathcal{Q}} \quad \frac{1}{2} \mathop{\mathbb{E}}_{D\sim\mathfrak{D}} \mathop{\mathbb{E}}_{S\sim D^n} \mathop{\mathbb{E}}_{(x,y)\sim D} \min\{(y - \langle A w_{\mathcal{Q}} + b, x\rangle)^2, 1\} \leq \frac{\sqrt{Tn}+1}{2\sigma T\sqrt{n}} \|w_{\mathcal{Q}}\|^2 \tag{4.16}$$
$$+ \frac{1}{2T\sqrt{n}} \sum_{t=1}^{T} \|(A_t - I_d) w_{\mathcal{Q}} + b_t\|^2 + \frac{1}{T} \sum_{t=1}^{T} \frac{1}{n} \sum_{i=1}^{n} (y_i^t - \langle A_t w_{\mathcal{Q}} + b_t, x_i^t\rangle)^2 + const.$$

The bounds (4.15) and (4.16) suggest learning a data-dependent hyperposterior by minimizing their right hand sides with respect to $w_{\mathcal{Q}}$. We will refer to the resulting algorithms as *Prior Learning with Gaussian hyperprior (PL-G)*.

In a practical implementation of PL-G for binary classification we replace $\bar{\Phi}$ in (4.15) by its convex relaxation, $\Phi_{cvx}(z) = \frac{1}{2} - \frac{z}{\sqrt{2\pi}}$, if $z \leq 0$ and $\Phi_{cvx}(z) = \bar{\Phi}(z)$ otherwise, and use the conjugate gradient method for finding the minimum. PL-G for regression problems is even simpler, because optimizing (4.16) has a closed form solution:

$$w_{\mathcal{Q}} = -\Big(D + \frac{\sqrt{Tn}+1}{\sigma T\sqrt{n}} I_d + \frac{1}{T\sqrt{n}} \sum_{t=1}^{T} A_t'^\top A_t'\Big)^{-1} \Big(c + \frac{1}{T\sqrt{n}} \sum_{t=1}^{T} A_t'^\top b_t\Big),$$
$$\text{where } A_t' = A_t - I_d,\ D = \frac{2}{T} \sum_{t=1}^{T} \frac{1}{n} A_t^\top X_t X_t^\top A_t, \tag{4.17}$$
$$c^\top = \frac{2}{T} \sum_{t=1}^{T} \frac{1}{n} \Big(\frac{C}{n} Y_t^\top X_t^\top A_t^\top X_t X_t^\top A_t - Y_t^\top X_t^\top A_t\Big).$$

Representation transfer techniques are based on the assumption that the weight vectors for different tasks lie in a low-dimensional subspace. In order to use Theorem 11 for learning such a subspace in a principled way we start with representing k-dimensional subspaces of $\mathbb{R}^d$ by $d \times k$ matrices with orthogonal columns, i.e. elements of the Stiefel manifold $V_{d,k}$. At the beginning of the learning process all subspaces are equally likely, thus we let the hyperprior $\mathcal{P}$ be the uniform distribution over $V_{d,k}$ [30]:

$$p_{\mathcal{P}}(B) = \frac{1}{C_0} \quad \text{for any } B \in V_{d,k}, \tag{4.18}$$

where $C_0 = {}_0F_1(\frac{1}{2}d, 0)$. For the hyperposterior $\mathcal{Q}$ we look for a distribution that concentrates its mass around a specific subspace, M. For this we employ a special case of Langevin distribution, $D(I_k, M)$, which can be interpreted as an analog of the Gaussian distribution on $V_{d,k}$:

$$p_{\mathcal{Q}}(B) = \frac{1}{C_1} \exp(\mathrm{tr}(M^\top B)) \quad \text{for any } B \in V_{d,k}, \tag{4.19}$$

where $C_1 = {}_0F_1(\frac{1}{2}d, \frac{1}{4}M^\top M)$. This distribution is parametrized by a $d \times k$ matrix M with $M^\top M = I_k$ that represents the most promising subspace.

As before we use Gaussian distributions for priors and posteriors, however now they are defined only within the subspaces sampled from $\mathcal{P}$ or $\mathcal{Q}$:

$$P = \mathcal{N}(0, \sigma I_k) \qquad \text{and} \qquad Q = \mathcal{N}(w_Q, \sigma I_k). \tag{4.20}$$

For the learning algorithm we select ridge regression that again is defined only within the subspace determined by the prior:

$$w_Q = \arg\min_w \left(\|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} (y_i - \langle w, B^\top x_i \rangle)^2 \right), \tag{4.21}$$

where B is the matrix representing the subspace and $B^\top x$ is the projected representation of the training data in this subspace.

Now we can compute the complexity terms in (4.10). First, note that for any $M \in V_{d,k}$ there exits an orthogonal matrix $L \in \mathbb{R}^{d \times d}$ such that $LM = J = \{\delta_{ij}\} \in \mathbb{R}^{d \times k}$. Therefore if $B \sim D(I_k, M)$, than $LB \sim D(I_k, LM) = D(I_k, J)$. Thus, the entropy of $D(I_k, M)$

is equal to the entropy of $D(I_k, J)$ and is a constant independent of M. Since $\mathcal{P}$ is uniform, $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$ depends only on the differential entropy of $\mathcal{Q}$ and thus is also a constant independent of M. Furthermore, $\mathrm{KL}(Q_t(S_t, P)\|P) = \frac{1}{2\sigma}\|w_t(B)\|^2$, where B is the representation of the selected subspace. Thus, we obtain the following corollary of Theorem 11:

$$\mathrm{er}(M) \leq \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{B\sim\mathcal{Q}} \left\{ \widehat{\mathrm{er}}(w_t(B)) + \frac{1}{2\sigma\sqrt{n}}\|w_t(B)\|^2 \right\} + const, \tag{4.22}$$

where $w_t(B) = \frac{C}{n}\left(I_k + \frac{C}{n}B^\top X_t X_t^\top B\right)^{-1} B^\top X_t Y_t$. Interpreting the right hand side as a quality measure, we can conclude that a representation, M, can be considered promising for future tasks, if for all observed tasks it allows for classification with small loss and small weight vector norm, i.e. large margin.

For a practical implementation for both binary classification and regression problems we use the standard squared loss, which is an upper bound on both the $0/1$ and the truncated squared losses. Moreover, we substitute all expectations over $\mathcal{Q}$ by their values at its mode, M, and replace the error of any Gibbs predictor by the error of the deterministic predictor defined by the mode of the posterior distribution, $w_t(M)$. As a result we obtain a quadratic optimization problem over the Stiefel manifold, which we solve using gradient descent with curvilinear search [98] and call the corresponding algorithm *Prior Learning with Langevin hyperprior (PL-L)*.

In order to examine the quality of priors learned by minimizing (4.15), (4.16) and (4.22), we evaluate PL-G and PL-L on three real world datasets:

- The *Land Mine Detection* [99] dataset consists of 14820 data points, each represented by 9 features extracted from radar images and associated with a binary label corresponding to landmine or clutter. Data is collected from 29 geographical regions and we treat each region as a binary classification task. We add a bias term, resulting in 10-dimensional features.

- *London School Data* is a regression dataset, containing exam scores of 15362 students from 139 schools, where each school constitutes a task. We use the same procedure as in [81] to encode 4 school-specific, 3 student-specific features and a year of examination in a set of binary features. With a bias term the resulting

dimensionality is $d = 28$. We divide labels (examination scores) by their maximum value, thus we can assume that the squared loss will not exceed 1. We report the squared error multiplied by the squared value of the maximum examination score.

- The *Animals with Attributes Dataset* [49] contains 30475 images from 50 classes. We use PCA to reduce original 2000-dimensional features to 100 dimensions, which we then l_2-normalize and add a bias term. We form 49 binary classification tasks, each of them is a binary classification of the largest class *collie* versus one of the remaining classes. To prevent data overlap between different tasks, for every task we use 2% of the data (approximately 20 images) available for *collie* class and the same amount of images from the negative class.

We compare PL-G and PL-L to ordinary ridge regression, adaptive ridge regression (ARR), i.e. Equation (4.11) with the prior w_{pr} set to the average of the weight vectors from the observed tasks, and with the ELLA algorithm [81] that learns a subspace representation using structured sparsity constraints, also with squared loss.

In order to examine the effect of different number of observed tasks on the performance of the algorithms on future tasks, in each experiment we set aside a subset of tasks as unobserved (9 in Landmines, 39 in Schools, 9 in Animals) and use different fractions of the remaining ones for training. To get reliable estimates of the transfer risk, we repeat the experiment 100 times for each dataset and report the mean errors and standard errors of the mean.

The only free parameter C in PL-G, PL-L and ARR we select from $\{10^{-3} \dots 10^{3}\}$ using the following modification of the ordinary 3-fold cross-validation: we split the data of each task into three parts and use the first third of all tasks jointly to learn a prior, then we train individual predictors on the second part of the data and test their quality on the last third. Using the same procedure, we select the regularization strength parameter μ of the ELLA algorithm, while the remaining parameters are fixed at their default values. We set the regularization parameter of the Ridge Regression using ordinary 3-fold cross-validation.

The results of the empirical evaluation are reported in Figure 4.1. For the Landmine dataset we report the value of area under the ROC curve, because the tasks are unbalanced. By construction problems on the Animals with Attributes dataset are

balanced, thus we report the standard mean classification error. Performance on the Schools dataset is measured using the mean squared error.

First, note that all the methods that use information transfer for sufficiently many observed tasks outperform the Ridge Regression baseline, indicating that the tasks are related in the considered experimental settings. Moreover, their performance improves with the number of observed tasks, which confirms the intuition that more observed tasks contain more information for the lifelong learning scenario.

In order to illustrate the effect of the hyperprior, we report performance of PL-G for two different values for the Gaussian hyperprior variance. We observe that higher variance ($\sigma = 10$) leads to faster convergence, compared to $\sigma = 1$, for which the adaption process is more conservative and many tasks are needed to find a reliable hyperposterior.

Overall, PL-G and PL-L are comparable to the existing, manually designed methods - ARR and ELLA, respectively. This indicates that the generalization bound of Theorem 11 can be used to derive principled and reasonably performing algorithms for lifelong learning. At the same time it provides an alternative view on the implicit assumptions of possible learning methods by reformulating them in terms of hyperpriors/hyperposterios.

4.1.2 Non-i.i.d. tasks

The original analysis of Baxter [14], as well as Theorems 10 and 11 rely on the assumption that the observed tasks, as well as new ones are sampled i.i.d. from the same task environment. This assumption makes it possible to rigorously argue about the future of the learning process based on the observations. However, it limits the applicability of these results in practice. In this subsection we will discuss two possible relaxations of the i.i.d. assumption that nevertheless still can be used to analyze the future of the lifelong learning process. These results were published in the paper "Lifelong Learning with Non-i.i.d. Tasks" [72].

The simplest and most intuitive way to relax the i.i.d. assumption is to keep the task environment fixed, but allow dependencies between the observed tasks. In this case tasks are identically, but not independently distributed. In terms of relatedness assumptions this setting is equivalent to the i.i.d. case, since in lifelong learning these

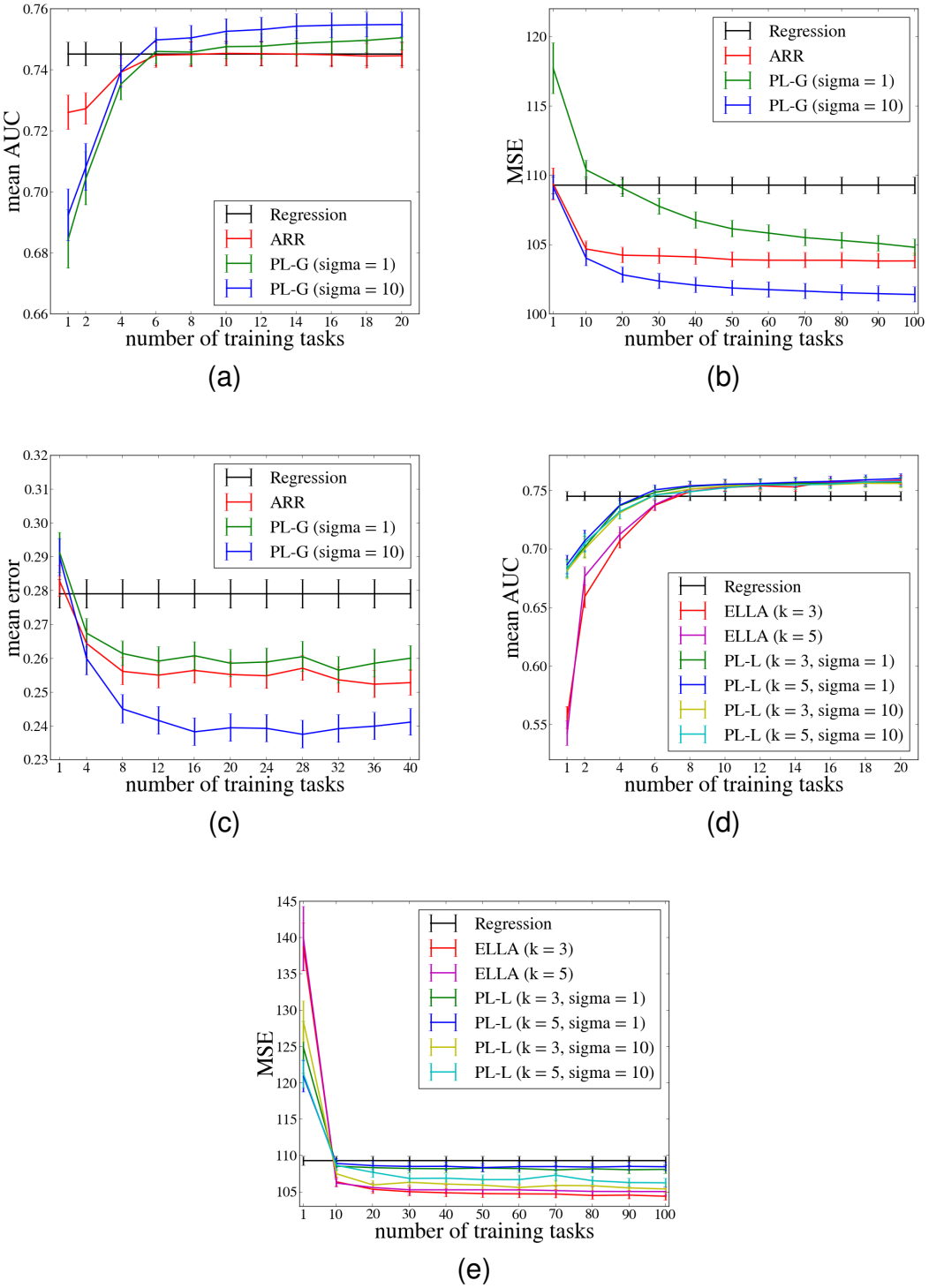


Figure 4.1: (a) mean AUC vs number of training tasks on Landmine dataset for ridge regression, ARR and PL-G with $\sigma = 1$ and $\sigma = 10$, (b) mean squared error vs number of training tasks on Schools dataset for ridge regression, ARR and PL-G with $\sigma = 1$ and $\sigma = 10$, (c) mean error vs number of training tasks on Animals dataset for ridge regression, ARR and PL-G with $\sigma = 1$ and $\sigma = 10$, (d) mean AUC vs number of training tasks for ELLA and PL-L with number of basis vectors $k = 3$ and $k = 5$ and variance $\sigma = 1$ and $\sigma = 10$ on Landmine dataset, (e) MSE vs number of training tasks for ELLA and PL-L with number of basis vectors $k = 3$ and $k = 5$ and variance $\sigma = 1$ and $\sigma = 10$ on Schools dataset.

assumptions are formulated on the task environment level and this is assumed to be constant. Thus, we can keep the same paradigm as that used for proving Theorem 11. As commonly done in analyzing lifelong learning, the proof of Theorem 11 consists of two steps. First, one bounds the difference between the empirical error $\widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q})$ and the average expected error over the observed tasks:

$$\mathrm{er}_{\mathbf{D}}(\mathcal{Q}) = \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{h\sim Q_t(P,S_t)} \mathop{\mathbb{E}}_{(x,y)\sim D_t} \ell(h(x),y). \tag{4.23}$$

This step is performed conditioned on the observed tasks. Thus, the corresponding training samples remain independent and one can keep this part of the proof without changes.

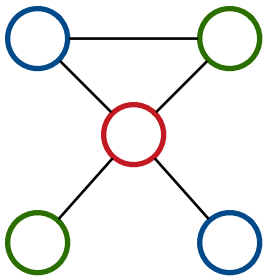
In contrast, the second step of the proof of Theorem 11 that bounds the difference between $\mathrm{er}_{\mathfrak{D}}(\mathcal{Q})$ and $\mathrm{er}_{\mathbf{D}}(\mathcal{Q})$ directly exploits the i.i.d. assumption on the observed tasks and therefore cannot be used. In order to extend this analysis one has to be able to quantify the amount of the dependencies between the observed tasks, because presumably it will effect the performance of the learner and the corresponding guarantees. In particular, if we imagine an extreme case, when the observed tasks are sampled by first randomly sampling a task from the environment and then repeating the obtained task $T-1$ times, it seems intuitive that such observation will contain only limited amount of information about the environment and thus the convergence rate of the upper bound should become slower compared to the i.i.d. case.

In order to formally quantify such effects we use the properties of a dependency graph built on the observed tasks.

Definition 7. *The dependency graph* $\Gamma(\mathbf{t}) = (V, E)$ of a set of random variables $\mathbf{t} = (t_1, \ldots, t_n)$ is such that:

- the set of vertices V equals $\{1, \ldots, n\}$,
- there is no edge between i and j if and only if t_i and t_j are independent.

Definition 8. Let $\Gamma = (V, E)$ be an undirected graph with $V = \{1, \ldots, n\}$. A set $\mathbf{C} = \{(C_j, w_j)\}_{j=1}^{k}$, where $C_j \subset V$ and $w_j \in [0, 1]$ for all j, is a *proper exact fractional cover* [94] if:



$$\mathbf{w}(\mathbf{C}) = k = 3$$

Figure 4.2: Illustration of the concept of a graph cover. Nodes with the same color correspond to the same subset in the cover.

- for every j all vertices in C_j are independent,
- $\cup_j C_j = V$,
- for every $i \in V \quad \sum_{j=1}^{k} w_j \mathbb{I}_{i \in C_j} = 1$.

The sum of the weights $\mathbf{w}(\mathbf{C}) = \sum_{j=1}^{k} w_j$ is called *the chromatic weight* of $\mathbf{C}$ and k is called *the size* of $\mathbf{C}$.

By adopting ideas from chromatic PAC-Bayesian bounds [77], we obtain the following generalization of Theorem 11:

Theorem 12 (Theorem 4 in [72])**.** *Let $\langle \mathfrak{T}, \mathfrak{D} \rangle$ be any task environment, $\mathcal{P}$ be any fixed hyper-prior distribution, Γ be the dependency graph of the observed T tasks and $\mathbf{C}$ be any proper exact fractional cover of Γ of size k. Then for any $\delta > 0$ the following holds with probability at least $1 - \delta$ uniformly for all hyper-posterior distributions $\mathcal{Q}$:*

$$\mathrm{er}_{\mathfrak{D}}(\mathcal{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q}) + \frac{1 + \sqrt{\mathbf{w}(\mathbf{C}) T n}}{T\sqrt{n}} \mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \frac{1}{T\sqrt{n}} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{P \sim \mathcal{Q}} \mathrm{KL}(Q_t(P, S_t)||P) + \frac{T + 8 \log 2/\delta}{8T\sqrt{n}} + \frac{\sqrt{\mathbf{w}(\mathbf{C})}(1 + 8 \log 2/\delta + 8 \log k)}{8\sqrt{T}}. \tag{4.24}$$

The form of inequality (4.24) is exactly the same as of inequality (4.10) for the i.i.d. case. The only difference is the presence of the weight of the fractional cover $\mathbf{C}$ in the terms, corresponding to the task environment level, i.e. those coming from bounding the difference between $\mathrm{er}_{\mathfrak{D}}$ and $\mathrm{er}_{\mathbf{D}}$. In particular, now these terms converge to zero as the

number of the observed tasks T grows as $\sqrt{\mathbf{w}(\mathbf{C})/T}$, compared to $1/\sqrt{T}$ for the i.i.d. case. Thus, the weight of the cover represents the amount of the dependencies among the observed tasks and can be used to quantify its effects on the convergence rate. The fastest convergence is obtained by using a cover with minimal chromatic weight, $\chi^*(\Gamma)$, which satisfies the following inequality [94]:

$$1 \leq c(\Gamma) \leq \chi^*(\Gamma) \leq \Delta(\Gamma) + 1, \tag{4.25}$$

where $c(\Gamma)$ is the order of the largest clique in Γ and $\Delta(\Gamma)$ is the maximum degree of a vertex in Γ. In the extreme case discussed before, where all the observed tasks are actually the same, the dependency graph is fully connected and its weight is T. This results in the corresponding terms not converging to zero, which confirms the intuition that observing the same problem again and again does not increase the knowledge about the whole task environment. In the opposite case, when the tasks are actually i.i.d., the weight is one, because the dependency graph contains no edges and inequality (4.24) transforms into (4.10). Thus Theorem 12 is a strict generalization of Theorem 11.

Next we discuss what can be learned when the observed tasks are not identically distributed. In particular, we consider a situation when the task environment is allowed to gradually change: every new task $t+1$ is sampled from a distribution $\mathfrak{D}_{t+1}$ over the tasks $\mathfrak{T}$ that can depend on the history of the process. In such setting the previously explored idea of automatically inferring a prior does not seem reasonable anymore, because such prior or, analogously, inductive bias in Baxter's works characterizes a task environment and models similarity between the tasks from the same environment. In contrast, in case of changing environments one would prefer to be able to model the change. For this we propose to learn a transfer algorithm that produces a solution for the current task based on the corresponding sample set and the data from the previous task. More formally, we assume that the learner has access to a set $\mathcal{A}$ of learning algorithms that produce a posterior distribution Q_{t+1} for task $t+1$ based on the training samples S_t and S_{t+1}, and its goal is to identify an algorithm in this set that can be successfully applied to the next observed task.

For a task t and an algorithm $A \in \mathcal{A}$ we can write the corresponding expected and

empirical errors:

$$\mathrm{er}_{D_t}(A) = \mathop{\mathbb{E}}_{h\sim Q_t} \mathop{\mathbb{E}}_{(x,y)\sim D_t} \ell(h(x), y), \quad \widehat{\mathrm{er}}_{S_t}(A) = \mathop{\mathbb{E}}_{h\sim Q_t} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i^t), y_i^t), \tag{4.26}$$

where $Q_t = A(S_t, S_{t-1})$. Thus, the goal of the learner can be reformulated as to find A that minimizes er_{T+1} given the history of the observed T tasks.

Note that if the task environment would change arbitrary over time, there would be no relevant information to the future in the observed tasks. Thus, one has to assume some regularities in the environmental changes in order to be able to benefit from transferring information from the previous tasks to the new ones. In particular, we assume that the expected success of every algorithm in the considered set $\mathcal{A}$ does not change over time. In other words, every $A \in \mathcal{A}$ is associated with a value $\mathrm{er}(A)$:

$$\mathop{\mathbb{E}}_{\{E_{t-1},E_t\}} [\, \mathrm{er}_t(A) \,|\, E_1, \ldots, E_{t-2} \,] = \mathrm{er}(A), \tag{4.27}$$

for every $i = 2, \ldots, T+1$, with $E_t = (\mathfrak{D}_t, D_t, S_t)$. Note that while the left hand side of (4.27) depends on t, $\mathrm{er}(A)$ does not. This indicates that the quality of every considered transfer algorithm is assumed to be the same throughout the learning process. Note also that this is a natural assumption in a sense that if the performance of every algorithm would be allowed to change over time arbitrary the learner could end up in a situation where an algorithm that perfectly works on the observed sequence of tasks shows poor performance on the new ones. However, one could argue that assumption (4.27) is too strong. It seems sufficient to assume that the algorithm that leads to (close to) optimum performance remains the same over time, while assumption (4.27) also implies that its (and every other algorithm's) expected error does not change.

In order to be able to identify an algorithm $A \in \mathcal{A}$ with minimal $\mathrm{er}(A)$ in a principled way, we develop an upper bound on $\mathrm{er}(A)$ that consists only of computable quantities and holds uniformly in A and thus can be used to guide the learner. For this we adopt the construction with hyperpriors and hyperposteriors: we define $\mathcal{P}$ as a hyperprior distribution over the set of possible algorithms $\mathcal{A}$ and let $\mathcal{Q}$ be a possibly data-dependent hyperposterior. The quality of the hyper-posterior $\mathcal{Q}$ and its empirical counterpart are

given by:

$$\mathrm{er}(\mathcal{Q}) = \mathop{\mathbb{E}}_{A\sim\mathcal{Q}} \mathrm{er}(A), \tag{4.28}$$

$$\widehat{\mathrm{er}}(\mathcal{Q}) = \mathop{\mathbb{E}}_{A\sim\mathcal{Q}} \frac{1}{T-1}\sum_{t=2}^{T} \widehat{\mathrm{er}}_{S_t}(A). \tag{4.29}$$

Now we can formulate the corresponding generalization bound:

Theorem 13 (Theorem 7 in [72])**.** *For any hyperprior distribution $\mathcal{P}$ and any $\delta > 0$ with probability at least $1-\delta$ the following holds uniformly for all $\mathcal{Q}$:*

$$\begin{aligned}\mathrm{er}(\mathcal{Q}) \leq \widehat{\mathrm{er}}(\mathcal{Q}) + \frac{\sqrt{(T-1)n}+1}{(T-1)\sqrt{n}}\,\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \frac{1}{(T-1)\sqrt{n}}\sum_{t=2}^{T}\mathop{\mathbb{E}}_{A\sim\mathcal{Q}}\mathrm{KL}(Q_t\|P_t)\\ + \frac{(T-1)+8\log 2/\delta}{8(T-1)\sqrt{n}} + \frac{1+2\log 2/\delta}{2\sqrt{T-1}},\end{aligned} \tag{4.30}$$

where $P_2, \ldots, P_T$ are some reference prior distributions that should not depend on the training sets of subsequent tasks.

As all previously discussed lifelong learning generalization bounds (given by Theorems 10, 11, 12), the right hand side of (4.30) contains two types of complexity terms - one corresponding to the level of the changes in the task environment and task-specific terms. The difference is that the first one converges as $1/\sqrt{T-1}$ because instead of individual tasks in this setting the learner operates on pairs of subsequent tasks.

In order to illustrate how inequality (4.30) can be used to learn a transfer algorithm we consider a toy example (Figure 4.3), where $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \{-1, 1\}$ and the change in the environment between two steps is due to a constant rotation by $\theta_0 = \frac{\pi}{6}$ of the feature space. To further simplify the problem we assume that every task environment contains only one task, i.e. every $\mathfrak{D}_t$ is a delta peak. We also assume that the learner uses linear classifiers, $h(x) = \mathrm{sign}\langle w, x\rangle$, and $0/1$-loss, $\ell(y_1, y_2) = [\![y_1 \neq y_2]\!]$, for solving every task. For the set of transfer algorithms we use a one-parameter family of algorithms A_α ($\alpha \in \mathbb{R}$) that given two sample sets S_{prev} and S_{cur}, first rotates S_{prev} by the angle α, and then trains a linear SVM on the union of both sets. An elementary calculation shows that condition (4.27) is fulfilled, thus we can use the bound (4.30) as a criterion to determine a beneficial angle.

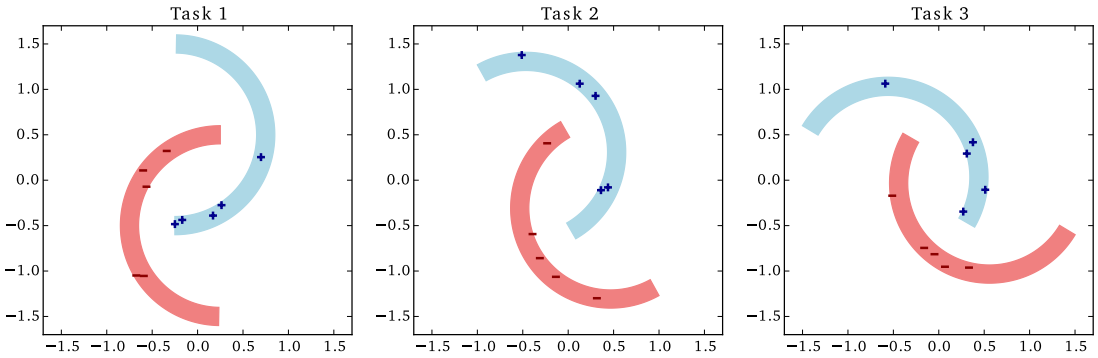


Figure 4.3: Illustration of three learning tasks sampled from a non-stationary environment. Shaded areas illustrate the data distribution, $+$ and $-$ indicate positive and negative training examples. Between subsequent tasks, the data distribution changes by a rotation. A transfer algorithm with access to two subsequent tasks can compensate for this by rotating the previous data into the new position, thereby obtaining more data samples to train on.

As before, we set posterior and prior distributions to be Gaussian:

$$Q_i = \mathcal{N}(w_i, I_2), \qquad P_i = \mathcal{N}(0, I_2). \tag{4.31}$$

Similarly, we set the hyperprior distribution $\mathcal{P}$ to be a zero-mean Gaussian, however, we increase its variance to 10 such that all reasonable angles α lie within one standard deviation from the mean. For the hyperposterior $\mathcal{Q}$ we select $\mathcal{N}(\theta, 1)$ and thus the goal is to determine the best θ. By plugging all these definitions into the bound (4.30) and applying the standard combination of tricks we obtain the following objective function:

$$\mathcal{J}(\theta) = \frac{\sqrt{(T-1)n}+1}{(T-1)\sqrt{n}} \cdot \frac{\theta^2}{20} + \frac{1}{T-1} \sum_{t=2}^{T} \left(\frac{\|w_t\|^2}{2\sqrt{n}} + \frac{1}{n} \sum_{i=1}^{n} \overline{\Phi}\left(\frac{y_i^t \langle w_t, x_i^t \rangle}{\|x_i^t\|} \right) \right). \tag{4.32}$$

By optimizing $\mathcal{J}(\theta)$ using $T = 2, \ldots, 11$ tasks with $n = 10$ samples each, one obtains rotation angles that on average lead to the test error of 14.2% for the $(T+1)$th task. As expected, this approach is more effective than not transferring any information from the previous tasks (which leads to the error of 15.0%), but not as beneficial as rotating always by the optimal angle of $\frac{\pi}{6}$ (which leads to the error of 13.5%).

4.2 Lifelong learning with weighted majority votes

Theorems 12 and 13 in the previous subsection show that the assumption that the observed tasks, as well as the future ones are sampled i.i.d. from a task environment can be relaxed. However, they still employ the notion of task environment. From the technical point of view, the concept of a task environment is a natural extension of the i.i.d. generative model used to analyze traditional single task learning. Essentially it just adds another level of randomness to the system and, as it is pointed out in the discussion in [14], this process can be continued by adding a third level and so on. On the other hand, motivation for lifelong learning often comes from the human education process and the human ability to exploit knowledge acquired from previously learned concepts for solving new problems more effectively. In this respect, modeling learning tasks as a random sample from an unknown distribution seems to be less natural. In particular, human education, which is often used to illustrate lifelong learning, is a highly organized process where new concepts are introduced gradually and their order is believed to optimize the effectiveness of the learning process. Presumably if students at school would study subjects in a random order, it would have negative effects. Thus, possibly, a task environment is not always the best way to model a lifelong learning system. Moreover, such a model results in a quality measure of the learner that is its performance *in expectation* over all tasks in the environment (see equation (4.1)) and the corresponding generalization bounds provide guarantees in terms of this quantity. However, in some situations good performance in expectation may not be sufficient. Imagine an autonomous robot that encounters different problems during the course of its life: failure on a single task may cause a robot to break and thus end up with, potentially expensive and time-consuming, repair. Finally, the theoretical analysis that employs Baxter's model relies on the assumption that at every time step the learner has access to all training examples for all previously observed tasks. This allows us to formulate a joint optimization problem. For example, Theorem 10 leads to a multi-task generalization of the standard empirical risk minimization approach. However, in practice it seems unlikely that an autonomous agent will be able to keep all this information. Therefore, there is clearly a need for an alternative *streaming* model of lifelong learning that (1) provides guarantees for every observed task, (2) does not make distributional assumptions on

the task generation process and (3) requires storage of only a compact description of the previous tasks, for example, only the corresponding learned hypotheses.

A few attempts have been made to progress in this direction. In domain adaptation it was previously argued for keeping only limited information from previously observed tasks [48]. A way to provide performance guarantees for every task in the multi-task setting was demonstrated in [18]. However these guarantees are a consequence of the employed relatedness assumption that implies that all the tasks have the same expected error. The first analysis of lifelong learning under streaming model was recently provided by Balcan *et al*. [11], where the authors concentrate on the case of learning linear classifiers. They propose an iterative algorithm that maintains a set of base predictors learned from previous tasks. For every new task it first attempts to learn it within the span of base predictors and, if that fails, learns a new linear predictor which is then added to the base set. Under the assumption that the tasks share a low-dimensional representation the authors demonstrate that the proposed method leads to sample complexity reductions compared to solving each task in isolation. However, their analysis relies also on the assumption that the marginal distributions for all tasks are isotropic log-concave and it is stated as an open problem whether this can be extended to other types of distributions. In this section we will discuss what kind of guarantees can be obtained for streaming lifelong learning with arbitrary marginal distributions. These results were published in "Lifelong Learning with Weighted Majority Votes" [75].

Our main insight is to consider weighted majority votes over the base predictors rather than their linear combinations. Apart from allowing us to consider any *ground* hypothesis set $\mathcal{H}$, as it will become evident later, this shift also introduces sufficient stability to the learned *base* set that allows exploiting it for later tasks and that cannot be achieved by linear combinations of linear predictors without additional assumptions on the marginal distributions. For a set of hypothesis $h_1, \ldots, h_k \in \mathcal{H}$ we define a set of weighted k-majority votes as:

$$\left\{ g : \mathcal{X} \to \mathcal{Y} \mid \exists w_1, \ldots, w_k \in \mathbb{R} : \; g(x) = \operatorname{sign}\left(\sum_{i=1}^{k} w_i h_i(x) \right) \right\} \tag{4.33}$$

and denote it by $\mathrm{MV}(h_1, \ldots, h_k)$.

We will focus on binary realizable case, thus every task encountered by the learner

can be represented as a pair $\langle D_i, h_i^* \rangle$ of a marginal distribution D_i over $\mathcal{X}$ and a deterministic labeling function $h_i^* : \mathcal{X} \to \{-1, 1\}$ that lies in some fixed ground class $\mathcal{H}$.

As in any transfer learning case, in the streaming lifelong learning setting potential benefits of the information transfer depend on how related the observed tasks are. A distinctive feature of the streaming case is that the learner has access only to the knowledge extracted from the previous tasks and thus the transfer is happening only in one direction. Therefore we will formulate the relatedness assumption in terms of a sequence of tasks, rather than a set (like in multi-task learning) or a task environment (like in Baxter's model). For this we will employ the following (pseudo-)metric over the hypothesis class with respect to a marginal distribution D:

$$d_D(h, h') = \mathop{\mathbb{E}}_{x \sim D} [\![h(x) \neq h'(x)]\!]. \tag{4.34}$$

Furthermore, we can define a distance from a hypothesis to a hypothesis space as

$$d_D(h, \mathcal{H}') = \min_{h' \in \mathcal{H}'} d_D(h, h') \tag{4.35}$$

and a distance between two sets of hypotheses as

$$d_D(\mathcal{H}, \mathcal{H}') = \max_{h \in \mathcal{H}} d_D(h, \mathcal{H}') = \max_{h \in \mathcal{H}} \min_{h' \in \mathcal{H}'} d_D(h, h'). \tag{4.36}$$

Though the latter is not necessarily a metric over subsets of the hypothesis space, it does satisfy the triangle inequality.

Now we can formulate the measures of task diversity that we will use:

Definition 9 (γ-separability)**.** A sequence of learning tasks $\langle D_1, h_1^* \rangle, \ldots, \langle D_n, h_n^* \rangle$ is *γ-separated*, if $d_{D_i}(h_i^*, MV(h_1^*, \ldots, h_{i-1}^*)) > \gamma$ for every i.

Definition 10 (γ-dimension)**.** A sequence of learning tasks $\langle D_1, h_1^* \rangle, \ldots, \langle D_n, h_n^* \rangle$ has *γ-effective dimension* k, if the largest γ-separated subsequence of these tasks has length k.

Definitions 9 and 10 are closely related to the corresponding notions employed in [11]: one obtains definitions from [11] by assuming the ground class $\mathcal{H}$ to be linear and

substituting weighted majority votes by linear combinations. However, this substitution leads to significant differences in the nature of the used complexity measures. In the case of linear predictors and their linear combinations small a dimensionality of the sequence is a relaxation of the often used assumption that the predictors for the tasks lie in a low-dimensional subspace. And, in addition, the order of the tasks is not important - shuffling the tasks does not increase the dimensionality of the sequence. In contrast, for weighted majority votes the order of tasks is of crucial importance. Indeed, the same set of tasks may have different dimensionality depending on how the tasks are ordered.

One intuitively advantageous scenario for a streaming lifelong learner is when throughout the course of learning most of the time information obtained from the observed tasks is sufficient for solving the current one. We formalize this intuition by saying that the γ-effective dimension k of the observed sequence of tasks is relatively small for a sufficiently small γ. This assumption is a relaxation of the relatedness assumption employed in [27], which states that there exist k hypotheses such that every task can be well explained by one of them.

In addition, we will need that the discrepancy between the marginal distributions of different tasks is small with respect to the ground hypothesis set $\mathcal{H}$:

$$\mathrm{disc}_{\mathcal{H}}(D_i, D_j) = \max_{h,h' \in \mathcal{H}} |d_{D_i}(h, h') - d_{D_j}(h, h')|. \tag{4.37}$$

Note that linear predictors together with an assumption that the marginal distributions for all tasks are isotropic log-concave, employed in [11], imply that the above discrepancy between any two tasks is zero. Since we do not make assumptions on the parametric form of the underlying data distributions, we need to control the discrepancy explicitly.

Algorithm 3 provides the pseudocode for the proposed procedure. It takes as input a set of 5 parameters: a fixed, *ground* hypothesis class $\mathcal{H}$, the total number of tasks in the sequence T, an upper bound on the γ-dimensionality of the task sequence k and accuracy and confidence parameters ϵ and δ. The algorithm maintains a set of *base* hypotheses $\tilde{h}_1, \ldots, \tilde{h}_{\tilde{k}}$. At the beginning this set is empty and the first task is solved within the ground class $\mathcal{H}$ using a training set S_1 large enough and the obtained hypothesis g_1 becomes the first element of the base set. After that for every new task the algorithm first attempts to solve it using a weighted majority vote over the set of

Algorithm 3 Lifelong learning of majority votes

Input parameters $\mathcal{H}, T, k, \epsilon, \delta$
set $\delta' = \delta/(2T)$, $\epsilon' = \epsilon/(8k)$
draw a training set S_1 from $\langle D_1, h_1^* \rangle$, s.t. $\Delta_1 := \Delta(\mathrm{VC}(\mathcal{H}), \delta', |S_1|) \leq \epsilon'$
$g_1 = \arg\min_{h \in \mathcal{H}} \mathcal{L}_{S_1}(h)$
set $\tilde{k} = 1$, $\tilde{h}_1 = g_1$
for $t = 2$ to T **do**
  draw a training set S_t from $\langle D_t, h_t^* \rangle$,
  s.t. $\Delta_t := \Delta(\mathrm{VC}(MV(\tilde{h}_1, \ldots, \tilde{h}_{\tilde{k}})), \delta', |S_t|) \leq \frac{\epsilon}{40}$
  $g_t = \arg\min_{h \in MV(\tilde{h}_1, \ldots, \tilde{h}_{\tilde{k}})} \widehat{\mathrm{er}}_{S_t}(h)$
  if $\widehat{\mathrm{er}}_{S_t}(g_t) + \sqrt{\widehat{\mathrm{er}}_{S_t}(g_t) \cdot \Delta_t} + \Delta_t > \epsilon$ **then**
    draw a training set S_t from $\langle D_t, h_t^* \rangle$, s.t. $\Delta_t := \Delta(\mathrm{VC}(\mathcal{H}), \delta', |S_t|) \leq \epsilon'$
    $g_t = \arg\min_{h \in \mathcal{H}} \widehat{\mathrm{er}}_{S_t}(h)$
    set $\tilde{k} = \tilde{k} + 1$, $\tilde{h}_{\tilde{k}} = g_i$
  end if
end for
return $g_1, \ldots, g_T$

base predictors collected so far. If it succeeds, it moves to the next task. Otherwise it learns the task using the ground hypothesis class and adds the resulting predictor to the base set.

An alternative way to look at the proposed method is to focus on the case when the ground class $\mathcal{H}$ is the set of linear predictors. Then the algorithm can be seen as a way to construct a neural network with $\mathrm{sign}()$ as the activation function. Indeed, in this case each base hypothesis is a node in a hidden middle layer or, in other words, a feature in the feature representation of the constructed neural net. When a new task is observed, it is either learned using the current representation, i.e. task-specific weights for the last layer are learned, or a new node is added to the middle layer and thus the feature representation is extended.

While this paradigm is very natural, the main challenge is to accurately select the internal parameters. In particular, one has to control the error propagation, since every task is solved based only on a finite training set. This is ensured using the results of Theorem 2 by choosing the training sets S_i large enough so that

$$\Delta_i := \Delta(\mathrm{VC}(\mathcal{H}_i), \delta', |S_i|) := C \frac{\mathrm{VC}(\mathcal{H}_i) \log(|S_i|) + \log(1/\delta')}{|S_i|}$$

is sufficiently small, where, depending on the situation, $\mathcal{H}_i$ is either the ground class $\mathcal{H}$ or the set of weighted majority votes over the current base set. In addition, it is important

to ensure that the algorithm does not search over the, potentially large, ground set $\mathcal{H}$ too often and thus may lead to sample complexity reductions over solving every task independently. In fact, the key component of the proof of the following theorem is to show that throughout the course of learning at most k tasks would not be solved as majority votes over earlier tasks:

Theorem 14 (Theorem 2 in [75])**.** *Consider running Algorithm 3 on a sequence of tasks with γ-effective dimension at most k and $\mathrm{disc}_{\mathcal{H}}(D_i, D_j) \leq \xi$ for all i, j. Then, if $\gamma \leq \epsilon/4$ and $k\xi < \epsilon/8$, with probability at least $1 - \delta$:*

- *The error of every task is bounded: $\mathrm{er}_{D_t, h_t^*}(g_t) \leq \epsilon$ for every $t = 1, \ldots, T$.*
- *The total number of labeled examples used is $\tilde{O}\left(\frac{Tk + \mathrm{VC}(\mathcal{H})k^2}{\epsilon}\right)$.*

Algorithm 4 Lifelong learning of majority votes with unkown horizon

Input parameters $\mathcal{H}, \epsilon, \delta$
set $\delta_1 = \delta/2$, $\epsilon_1' = \epsilon/16$
draw a training set S_1 from $\langle D_1, h_1^* \rangle$, s.t. $\Delta(\mathrm{VC}(\mathcal{H}), \delta_1, \|S_1\|) \leq \epsilon_1'$
$g_1 = \arg\min_{h \in \mathcal{H}} \widehat{\mathrm{er}}_{S_1}(h)$
set $\tilde{k} = 1$, $\tilde{h}_1 = g_1$
for $t = 2$ to T **do**
  set $l = \lfloor \log t \rfloor$, $m = \lfloor \log \tilde{k} + 1 \rfloor$
  set $\delta_t = \frac{\delta}{2^{2l+2}}$, $\epsilon_t' = \frac{\epsilon}{2^{2m+4}}$
  draw a training set S_t from $\langle D_t, h_t^* \rangle$, s.t. $\Delta(\mathrm{VC}(MV(\tilde{h}_1, \ldots, \tilde{h}_{\tilde{k}})), \delta_t, \|S_t\|) \leq \epsilon/40$
  $g_t = \arg\min_{h \in MV(\tilde{h}_1, \ldots, \tilde{h}_{\tilde{k}})} \widehat{\mathrm{er}}_{S_t}(h)$
  if $\widehat{\mathrm{er}}_{S_t}(g_t) + \sqrt{\widehat{\mathrm{er}}_{S_t}(g_t) \cdot \Delta} + \Delta > \epsilon$ **then**
    draw a training set S_t from $\langle D_t, h_t^* \rangle$, s.t. $\Delta(\mathrm{VC}(\mathcal{H}), \delta_t, \|S_t\|) \leq \epsilon_t'$
    $g_t = \arg\min_{h \in \mathcal{H}} \widehat{\mathrm{er}}_{S_t}(h)$
    set $\tilde{k} = \tilde{k} + 1$, $\tilde{h}_{\tilde{k}} = g_t$
  end if
end for
return $g_1, \ldots, g_T$

An important aspect of Algorithm 3 is that its internal parameters depend on the task horizon T and complexity k. In practice, however, one would expect both of these parameters to be unknown to the learner. By slightly modifying the method (see Algorithm 4) using the doubling trick one can avoid this dependence at the price of worse sample complexity guarantees that are summarized in the following theorem:

Theorem 15 (Theorem 3 in [75])**.** *Consider running Algorithm 4 on a sequence of tasks with γ-effective dimension at most k and $\mathrm{disc}_{\mathcal{H}}(D_i, D_j) \leq \xi$ for all i, j. Then, if $\gamma \leq \epsilon/4$ and $k\xi < \epsilon/8$, with probability at least $1 - \delta$:*

- *The error of every task is bounded:* $\mathrm{er}_{D_t,h_t^*}(g_t) \leq \epsilon$ *for every* $t = 1, \ldots, T$*.*
- *The total number of labeled examples used is* $\tilde{O}\left(\frac{Tk+\mathrm{VC}(\mathcal{H})k^3}{\epsilon}\right)$*.*

Theorems 14 and 15 quantify the intuition that if most tasks in a sequence are well learnable by weighted majority votes over the previously observed tasks, then the proposed streaming lifelong learning method will lead to sample complexity reductions. Indeed, learning every task independently using the ground class has the total sample complexity

$$\tilde{O}\left(\frac{\mathrm{VC}(\mathcal{H})T}{\epsilon}\right), \tag{4.38}$$

since by the assumption every task is realizable by the ground class. In contrast, the complexities of Algorithms 3 and 4 are $\tilde{O}\left(\frac{Tk+\mathrm{VC}(\mathcal{H})k^2}{\epsilon}\right)$ and $\tilde{O}\left(\frac{Tk+\mathrm{VC}(\mathcal{H})k^3}{\epsilon}\right)$ respectively, which is significantly smaller than (4.38) whenever the effective dimension k of the task sequence is much smaller than the total number of tasks T and the VC-dimension of the ground class. Thus the key condition for the sample complexity improvement is a small effective dimension of the observed sequence. Note, however, that both Algorithm 3 and 4 are in general not computationally efficient as they require performing ERM for every observed task.

4.3 Conclusion

In this chapter we have discuss the lifelong learning scenario which can be seen as an extension of multi-task learning with an additional source of uncertainty that comes from the fact that the learner does not know what tasks he will encounter in the future. We have illustrated this by extending the result for multiple kernel learning in the multi-task scenario (Theorem 5) to lifelong learning. Then we have shown how the original ideas of Baxter [14] can be translated from the PAC to PAC-Bayesian language, which results in Theorem 11 that can be used to develop principled lifelong learning algorithms. We have illustrated this process for parameter and representation transfer approaches, which resulted in PL-G and PL-L methods. The rest of the chapter was dedicated to investigation of possible relaxations on the core assumption of many lifelong learning analyses, which states that the observed tasks, as well as the future ones are sampled

i.i.d. from some task environment. First, under the assumption that the observed tasks are identically, but not independently distributed, we have shown how the amount of the dependencies between them influences the generalization guarantees. Second, we have shown that when the task environment is changing over time, but in a restricted way, it is possible to learn a transfer method that would compensate for the environmental changes. Lastly, in Section 4.2 we have discussed a more challenging streaming model of lifelong learning that does not make any assumptions on the task generation process.

5 Future directions

The key aspect of all transfer learning approaches, including multi-task and lifelong learning, are assumptions on the relatedness between the prediction tasks. Therefore it is important to identify and explore different kinds of relatedness that could make the information transfer beneficial. Most of the transfer methods concentrate on learning a "good" representation, where quality of a representation is measured by its approximation ability and simplicity. Such assumptions were exploited not only for linear classifiers, as was discussed in Section 3.1, but also for neural networks [52], where shared representation takes a form of a shared layers of a deep architecture. However, this is not the only option. One could also think of a good representation as one that leads to "easy" learning of every task, i.e. learning with fast rates. In this case one could use measures of distribution "niceness" that have been shown to lead to faster convergence rates, like Probabilistic Lipschitzness [95], as a quality measure of a learned representation. A completely different approach to defining task relatedness was explored in [3]. There the authors consider learning several tasks using deep neural networks under the assumption that what makes the tasks of interest similar is not the architecture of the network, but an optimization procedure and the proposed method aims at in inferring optimal gradient step updates for training neural networks corresponding to different tasks.

Probably a more general question is whether the existing models for multi-task and lifelong learning are relevant. In particular, multi-task learning is almost always defined as a setting where the learner is given a collection of annotated training sets for several tasks and the goal is to minimize the average expected error over all of them. However, as was discussed in Section 3.3 there could be some situations where not for

all tasks of interest labeled data is available. Or, as was pointed out in [65], the average performance might not be what one should care about. Lifelong learning is even less well-defined area and thus it is important to identify scenarios under which information transfer could be beneficial. Presumably, all these aspects of transfer learning depend on the area of application. Therefore it is important to put a greater focus on modeling problems that are considered by machine learning practitioners.

Bibliography

[1] Arvind Agarwal, Samuel Gerber, and Hal III Daume. Learning multiple tasks using manifold regularization. In *Conference on Neural Information Processing Systems (NIPS)*. 2010.

[2] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research (JMLR)*, 2005.

[3] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Conference on Neural Information Processing Systems (NIPS)*, 2016.

[4] Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[5] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Conference on Neural Information Processing Systems (NIPS)*, 2007.

[6] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 2008.

[7] Andreas Argyriou, Andreas Maurer, and Massimiliano Pontil. An algorithm for transfer learning in a heterogeneous environment. In *European Conference on Machine Learning (ECML)*, 2008.

[8] Andreas Argyriou, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. A spectral regularization framework for multi-task structure learning. In *Conference on Neural Information Processing Systems (NIPS)*, 2007.

[9] Sohail Bahmani, Bhiksha Raj, and Petros T. Boufounos. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research (JMLR)*, 2013.

[10] Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research (JMLR)*, 2003.

[11] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Efficient representations for lifelong learning and autoencoding. In *Workshop on Computational Learning Theory (COLT)*, 2015.

[12] Peter L. Bartlett, Sanjeev R. Kulkarni, and Steven E. Posner. Covering numbers for real-valued function classes. *IEEE transactions on information theory*, 1997.

[13] Jonathan Baxter. Learning model bias. In *Advances in Neural Information Processing Systems*, 1996.

[14] Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 2000.

[15] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 2008.

[16] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 2010.

[17] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. 2006.

[18] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Workshop on Computational Learning Theory (COLT)*, 2003.

[19] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision (ECCV)*, 2010.

[20] Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, 2005.

[21] Rich Caruana. Multitask learning. *Machine Learning*, 1997.

[22] Olivier Catoni. *PAC-Bayesian Supervised Classification (The Thermodynamics of Statistical Learning)*. Monograph Series of the Institute of Mathematical Statistics. IMS, 2007.

[23] Giovanni Cavallanti, Nicolò Cesa-Bianchi, and Claudio Gentile. Linear algorithms for online multitask classification. *Journal of Machine Learning Research (JMLR)*, 2010.

[24] Xinghua Lou Christian Widmer, Marius Kloft and Gunnar Ratsch. Regularization-based multitask learning: With applications to genome biology and biomedical imaging. *German Journal on Artificial Intelligence*, 2013.

[25] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[26] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *International Conference on Machine Learning (ICML)*, 2010.

[27] Koby Crammer and Yishay Mansour. Learning multiple tasks using shared hypotheses. In *Conference on Neural Information Processing Systems (NIPS)*, 2012.

[28] Monroe. D. Donsker and S. R. Srinivasa Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. I. *Communications on Pure and Applied Mathematics*, 1975.

[29] Joseph L Doob. Regularity properties of certain families of chance variables. *Transactions of the American Mathematical Society*, 1940.

[30] Thomas D. Downs. Orientation statistics. *Biometrika*, 1972.

[31] Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research (JMLR)*, 2005.

[32] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[33] Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *International Conference on Machine Learning (ICML)*, 2009.

[34] Mehmet Gönen, Melih Kandemir, and Samuel Kaski. Multitask learning using regularized multiple kernel learning. In *International Conference on Neural Information Processing (ICONIP)*, 2011.

[35] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 1992.

[36] Tom Heskes. Empirical bayes for learning to learn. In *International Conference on Machine Learning (ICML)*, 2000.

[37] Jerry L. Hintze and Ray D. Nelson. Violin plots: A box plot-density trace synergism. *The American Statistician*, 1998.

[38] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963.

[39] Tony Jebara. Multi-task feature and kernel selection for svms. In *International Conference on Machine Learning (ICML)*, 2004.

[40] Tony Jebara. Multitask sparsity via maximum entropy discrimination. *Journal of Machine Learning Research (JMLR)*, 2011.

[41] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[42] Wolf Kienzle and Kumar Chellapilla. Personalized handwriting recognition via biased regularization. In *International Conference on Machine Learning (ICML)*, 2006.

[43] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *International Conference on Very Large Data Bases (VLDB)*, 2004.

[44] Marius Kloft and Gilles Blanchard. On the convergence rate of ℓ_p-norm multiple kernel learning. *Journal of Machine Learning Research (JMLR)*, 2012.

[45] Mladen Kolar, John Lafferty, and Larry Wasserman. Union support recovery in multi-task learning. *Journal of Machine Learning Research (JMLR)*, 2011.

[46] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image Search with Relative Attribute Feedback. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[47] Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning (ICML)*, 2012.

[48] Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning (ICML)*, 2013.

[49] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2013.

[50] John Langford and John Shawe-Taylor. PAC-Bayes and margins. In *Conference on Neural Information Processing Systems (NIPS)*, 2002.

[51] François Laviolette and Mario Marchand. PAC-Bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers. *Journal of Machine Learning Research (JMLR)*, 2007.

[52] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, 2015.

[53] Karim Lounici, Massimilano Pontil, Alexandre B. Tsybakov, and Sara van De Geer. Taking advantage of sparsity in multi-task learning. In *Workshop on Computational Learning Theory (COLT)*, 2009.

[54] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Workshop on Computational Learning Theory (COLT)*, 2009.

[55] Andreas Maurer. Bounds for linear multi-task learning. *Journal of Machine Learning Research (JMLR)*, 2006.

[56] Andreas Maurer. Concentration inequalities for functions of independent variables. *Random Structures and Algorithms*, 2006.

[57] Andreas Maurer. Transfer bounds for linear feature learning. *Machine Learning*, 2009.

[58] Andreas Maurer and Massimiliano Pontil. Excess risk bounds for multitask learning with trace norm regularization. In *Workshop on Computational Learning Theory (COLT)*, 2013.

[59] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. In *International Conference on Machine Learning (ICML)*, 2013.

[60] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. An inequality with applications to structured sparsity and multitask dictionary learning. In *Workshop on Computational Learning Theory (COLT)*, 2014.

[61] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. arXiv:1505.06279, 2015.

[62] David A. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 1999.

[63] David A. McAllester. Simplified PAC-Bayesian margin bounds. In *Workshop on Computational Learning Theory (COLT)*, 2003.

[64] Colin McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*, 1989.

[65] Nishant A. Mehta, Dongryeol Lee, and Alexander G. Gray. Minimax Multi-Task Learning and a Generalized Loss-Compositional Paradigm for MTL. In *Conference on Neural Information Processing Systems (NIPS)*, 2012.

[66] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[67] Christos H. Papadimitriou. Worst-case and probabilistic analysis of a geometric location problem. *SIAM Journal on Computing*, 1981.

[68] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 2009.

[69] Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. PAC-Bayes bounds with data dependent priors. *Journal of Machine Learning Research (JMLR)*, 2012.

[70] Anastasia Pentina and Shai Ben-David. Multi-task and lifelong learning of kernels. In *Algorithmic Learning Theory (ALT)*, 2015.

[71] Anastasia Pentina and Christoph H. Lampert. A PAC-Bayesian bound for lifelong learning. In *International Conference on Machine Learning (ICML)*, 2014.

[72] Anastasia Pentina and Christoph H. Lampert. Lifelong learning with non-i.i.d. tasks. In *Conference on Neural Information Processing Systems (NIPS)*, 2015.

[73] Anastasia Pentina and Christoph H. Lampert. Active Task Selection for Multi-Task Learning. arXiv:1602.06518, 2016.

[74] Anastasia Pentina, Viktoriia Sharmanska, and Christoph H. Lampert. Curriculum learning of multiple tasks. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[75] Anastasia Pentina and Ruth Urner. Lifelong learning with weighted majority votes. In *Conference on Neural Information Processing Systems (NIPS)*, 2016.

[76] Alain Rakotomamonjy, Rémi Flamary, Gilles Gasso, and Stéphane Canu. ℓ_p-ℓ_q penalty for sparse linear and sparse multiple kernel multi-task learning. *IEEE Transactions on Neural Networks*, 2011.

[77] Liva Ralaivola, Marie Szafranski, and Guillaume Stempfel. Chromatic PAC-Bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes. *Journal of Machine Learning Research (JMLR)*, 2010.

[78] Bernardino Romera-Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. Exploiting unrelated tasks in multi-task learning. In *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2012.

[79] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[80] Paul Ruvolo and Eric Eaton. Active Task Selection for Lifelong Machine Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2013.

[81] Paul Ruvolo and Eric Eaton. ELLA: An efficient lifelong learning algorithm. In *International Conference on Machine Learning (ICML)*, 2013.

[82] Paul Ruvolo and Eric Eaton. Online multi-task learning via sparse dictionary optimization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[83] Wojciech Samek, Alexander Binder, and Motoaki Kawanabe. Multi-task learning via non-sparse multiple kernel learning. In *Computer Analysis of Images and Patterns*. 2011.

[84] Matthias W. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalization Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, 2003.

[85] Yevgeny Seldin, François Laviolette, Nicolò Cesa-Bianchi, John Shawe-Taylor, and Peter Auer. PAC-Bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 2012.

[86] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[87] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Learning to transfer privileged information. arXiv:1410.0389 [cs.CV], 2014.

[88] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[89] Nathan Srebro and Shai Ben-David. Learning bounds for support vector machines with learned kernels. In *Workshop on Computational Learning Theory (COLT)*, 2006.

[90] Dimitris Stamos, Samuele Martelli, Moin Nabi, Andrew McDonald, Vittorio Murino, and Massimiliano Pontil. Learning with dataset bias in latent subcategory models. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[91] Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 1995.

[92] Sebastian Thrun and Joseph O'Sullivan. Clustering learning tasks and the selective cross-task transfer of knowledge. Technical report, Carnegie-Mellon University.Computer science. Pittsburgh (PA US), 1995.

[93] Ilya Tolstikhin, Gilles Blanchard, and Marius Kloft. Localized complexities for transductive learning. In *Workshop on Computational Learning Theory (COLT)*, 2014.

[94] Daniel Ullman and Edward Scheinerman. *Fractional Graph Theory: A Rational Approach to the Theory of Graphs*. Wiley Interscience Series in Discrete Mathematics, 1997.

[95] Ruth Urner and Shai Ben-David. Probabilistic Lipschitzness: A niceness assumption for deterministic labels. In *Learning Faster from Easy Data - Workshop @ NIPS*, 2013.

[96] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 1984.

[97] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 1971.

[98] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 2013.

[99] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research (JMLR)*, 2007.

[100] Yu Zhang and Dit-Yan Yeung. Multi-task learning in heterogeneous feature spaces. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2011.

[101] Yang Zhou, Rong Jin, and Steven C.H. Hoi. Exclusive lasso for multi-task feature selection. In *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2010.

A Proofs of theorems in Chapter 3

A.1 Proof of Theorem 5

In order to prove Theorem 5 we employ the technique of covering numbers:

Definition 11. A subset $\tilde{A} \subset A$ is called an ϵ-*cover* of A with respect to a distance measure d, if for every $a \in A$ there exists a $\tilde{a} \in \tilde{A}$ such that $d(a, \tilde{a}) < \epsilon$. The *covering number* $N_d(A, \epsilon)$ is the size of the smallest ϵ-cover of A.

In particular, we will use covers of $\mathbb{H}^T = \cup_{K \in \mathcal{K}} \mathcal{H}_K^T$ with respect to ℓ_∞ metric associated with a sample $\mathbf{x} \in \mathcal{X}^{(T,n)}$:

$$d_\infty^{\mathbf{x}}(\mathbf{h}, \mathbf{h}') = \max_{t=1\ldots T} \max_{i=1\ldots n} |h_t(x_i^t) - h_t'(x_i^t)|. \tag{A.1}$$

The corresponding uniform covering number $N_{(T,n)}(\mathbb{H}^T, \epsilon)$ is given by considering the maximum covering number over all possible samples $\mathbf{x} \in \mathcal{X}^{(T,n)}$:

$$N_{(T,n)}(\mathbb{H}^T, \epsilon) = \max_{\mathbf{x} \in \mathcal{X}^{(T,n)}} N_{d_\infty^{\mathbf{x}}}(\mathbb{H}^T, \epsilon). \tag{A.2}$$

Using this notion we can obtain the following result:

Theorem 16. *For any $\epsilon, \gamma > 0$, if $n > 2/\epsilon^2$, we have that:*

$$\Pr\left\{\exists \mathbf{h} \in \mathbb{H}^T : \ \mathrm{er}_{\mathbf{D}}(\mathbf{h}) > \widehat{\mathrm{er}}_{\mathbf{S}}^{\gamma}(\mathbf{h}) + \epsilon\right\} \leq 2N_{(T,2n)}(\mathbb{H}^T, \gamma/2) \exp\left(-\frac{Tn\epsilon^2}{8}\right). \tag{A.3}$$

Proof. We utilize the standard 3-steps procedure (see Theorem 10.1 in [4]). If we denote:

$$\begin{aligned} Q &= \left\{\mathbf{S} \in (\mathcal{X} \times \mathcal{Y})^{(T,n)} : \ \exists \mathbf{h} \in \mathbb{H}^T : \ \ \mathrm{er}_{\mathbf{S}}(\mathbf{h}) > \widehat{\mathrm{er}}_{\mathbf{S}}^{\gamma}(\mathbf{h}) + \epsilon\right\} \\ R &= \left\{\mathbf{S} = (\mathbf{S_1}, \mathbf{S_2}) \in (\mathcal{X} \times \mathcal{Y})^{(T,n)} \times (\mathcal{X} \times \mathcal{Y})^{(T,n)} : \right. \\ &\qquad \left. \exists \mathbf{h} \in \mathbb{H}^T : \ \ \widehat{\mathrm{er}}_{\mathbf{S_2}}(\mathbf{h}) > \widehat{\mathrm{er}}_{\mathbf{S_1}}^{\gamma}(\mathbf{h}) + \epsilon/2\right\}, \end{aligned}$$

then according to the symmetrization argument $\Pr(Q) \leq 2\Pr(R)$. Therefore, instead of bounding the probability of Q, we can bound the probability of R.

Let Γ_{2n} be a set of permutations σ on the set $\{(1,1),\dots,(T,2n)\}$ such that $\{\sigma(t,i),\sigma(t,n+i)\}=\{(t,i),(t,n+i)\}$ for every $t=1,\dots,T$ and $i=1,\dots,n$. Then $\Pr(R)\le\max_{\mathbf{S}\in(\mathcal{X}\times\mathcal{Y})^{(T,2n)}}\Pr_\sigma(\sigma\mathbf{S}\in R)$.

The last step is a reduction to a finite class. Fix some $\mathbf{S}\in(\mathcal{X}\times\mathcal{Y})^{(T,2n)}$ and let σ be a permutation such that $\sigma\mathbf{S}\in R$. By definition there exists $\mathbf{h}\in\mathbb{H}^T$ such that $\widehat{\mathrm{er}}_{\mathbf{S_2}}(\mathbf{h})>\widehat{\mathrm{er}}^{\gamma}_{\mathbf{S_1}}(\mathbf{h})+\epsilon/2$, where $(\mathbf{S_1},\mathbf{S_2})=\sigma\mathbf{S}$:

$$\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=n+1}^{2n}[\![h_t(x^t_{\sigma(ti)})y^t_{\sigma(ti)}<0]\!]>\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}[\![h_t(x^t_{\sigma(ti)})y^t_{\sigma(ti)}<\gamma]\!]+\epsilon/2.$$

Denote by $\mathbf{x}=(x^t_i)\in X^{(T,2n)}$ the $\mathcal{X}$-part of sample $\mathbf{S}$ and let Γ be a $\gamma/2$-cover of $\mathbb{H}^T$ with respect to $d^{\mathbf{x}}_\infty$. If we denote by $\tilde{\mathbf{h}}$ the function in the cover Γ corresponding to $\mathbf{h}$, then the following inequalities hold:

$$\text{if } \tilde{h}_t(x^t_i)y^t_i<\frac{\gamma}{2},\ \text{then } h_t(x^t_i)y^t_i<\gamma$$
$$\text{if } h_t(x^t_i)y^t_i<0,\ \text{then } \tilde{h}_t(x^t_i)y^t_i<\frac{\gamma}{2}.$$

By combining them with the previous inequality we obtain that:

$$\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=n+1}^{2n}\left[\!\!\left[\tilde{h}_t(x^t_{\sigma(ti)})y^t_{\sigma(ti)}<\frac{\gamma}{2}\right]\!\!\right]>\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}\left[\!\!\left[\tilde{h}_t(x^t_{\sigma(ti)})y^t_{\sigma(ti)}<\frac{\gamma}{2}\right]\!\!\right]+\frac{\epsilon}{2}.$$

Now, if we define the following indicator: $v(\tilde{\mathbf{h}},t,i)=[\![\tilde{h}_t(x^t_i)y^t_i<\gamma/2]\!]$, then:

$$\begin{aligned}\Pr_\sigma\{\sigma\mathbf{S}\in R\}&\le\Pr_\sigma\left\{\exists\tilde{\mathbf{h}}\in\Gamma:\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}(v(\tilde{\mathbf{h}},\sigma(t,n+i))-v(\tilde{\mathbf{h}},\sigma(i,i)))>\frac{\epsilon}{2}\right\}\\&\le|\Gamma|\max_{\tilde{\mathbf{h}}\in\Gamma}\Pr_\sigma\left\{\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}(v(\tilde{\mathbf{h}},\sigma(t,n+i))-v(\tilde{\mathbf{h}},\sigma(t,i)))>\frac{\epsilon}{2}\right\}\end{aligned}$$

$$=|\Gamma|\max_{\tilde{\mathbf{h}}\in\Gamma}\Pr_\beta\left\{\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n}\sum_{i=1}^{n}|v(\tilde{\mathbf{h}},t,n+i)-v(\tilde{\mathbf{h}},t,i)|\beta_{ti}>\frac{\epsilon}{2}\right\}=(*),$$

where β_{ti} are independent random variables uniformly distributed over $\{-1,+1\}$. Then $\{|v(\tilde{\mathbf{h}},i,m+j)-v(\tilde{\mathbf{h}},i,j)|\beta_{ij}\}$ are Tn independent random variables that take values

between -1 and 1 and have zero mean. Therefore by Hoeffding's inequality:

$$(*) \leq |\Gamma| \exp\left(-\frac{2(Tn)^2\epsilon^2/4}{Tn \cdot 4}\right) = |T| \exp\left(-\frac{Tn\epsilon^2}{8}\right).$$

Since by the definition $|\Gamma| \leq N_{(T,2n)}(\mathbb{H}^T, \gamma/2)$, we conclude the proof of Theorem 16. □

By using the same technique as for proving Theorem 16, we can obtain a lower bound on the difference between the empirical error rate $\widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathbf{h})$ and the expected error rate with double margin:

$$\mathrm{er}^{2\gamma}_{\mathbf{D}}(\mathbf{h}) = \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{(x,y)\sim D_t} [\![yh_t(x) < 2\gamma]\!]. \tag{A.4}$$

Theorem 17. *For any $\epsilon > 0$, if $n > 2/\epsilon^2$, the following holds:*

$$\Pr\left\{\exists \mathbf{h} \in \mathbb{H}^T : \mathrm{er}^{2\gamma}_{\mathbf{D}}(\mathbf{h}) < \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathbf{h}) - \epsilon\right\} \leq 2N_{(T,2n)}(\mathbb{H}^T, \gamma/2) \exp\left(-\frac{Tn\epsilon^2}{8}\right). \tag{A.5}$$

The last step in proving Theorem 5 is to upper bound the covering numbers, used in Theorems 16 and 17, in terms of the pseudodimension of the kernel family $\mathcal{K}$:

Lemma 1. *For any set $\mathcal{K}$ of kernels bounded by B^2 ($K(x,x) \leq B^2$ for all $K \in \mathcal{K}$ and all $x \in \mathcal{X}$) with pseudodimension d_ϕ the following inequality holds:*

$$N_{(T,n)}(\mathbb{H}^T, \epsilon) \leq 2^T \left(\frac{4eT^2n^3B^2}{\epsilon^2 d_\phi}\right)^{d_\phi} \left(\frac{16nB^2}{\epsilon^2}\right)^{\frac{64B^2T}{\epsilon^2}\log\left(\frac{e\epsilon n}{8B}\right)}.$$

In order to prove this result, we first introduce some additional notation. For a sample $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ we define l_∞ distance between two functions as:

$$d^{\mathbf{x}}_\infty(f_1, f_2) = \max_{i=1\ldots n} |f_1(x_i) - f_2(x_i)|. \tag{A.6}$$

Then the corresponding uniform covering number is:

$$N_n(\mathcal{F}, \epsilon) = \sup_{\mathbf{x}\in\mathcal{X}^n} N_{d^{\mathbf{x}}_\infty}(\mathcal{F}, \epsilon) \tag{A.7}$$

We also define l_∞ distance between kernels with respect to a sample $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_T}) \in \mathcal{X}^{(T,n)}$ as:

$$D_\infty^{\mathbf{x}}(K, \hat{K}) = \max_t |K_{\mathbf{x_t}} - \hat{K}_{\mathbf{x_t}}|_\infty \tag{A.8}$$

where $K_{\mathbf{x}}$ is a kernel matrix associated with a sample $\mathbf{x}$. The corresponding uniform covering number is:

$$N_{(T,n)}(\mathcal{K}, \epsilon) = \sup_{\mathbf{x} \in X^{(T,n)}} N_{D_\infty^{\mathbf{x}}}(\mathcal{K}, \epsilon).$$

In contrast, in [89] the distance between two kernels is defined based on a single sample $\mathbf{x} = (x_1, \ldots, x_n)$ of size n:

$$D_\infty^{\mathbf{x}}(K, \hat{K}) = |K_{\mathbf{x}} - \hat{K}_{\mathbf{x}}|_\infty \tag{A.9}$$

and the corresponding covering number is $N_n(\mathcal{K}, \epsilon)$. Note that this definition is in strong relation with ours: $N_{(T,n)}(\mathcal{K}, \epsilon) \leq N_{Tn}(\mathcal{K}, \epsilon)$, and therefore, by Lemma 3 in [89]:

$$N_{(T,n)}(\mathcal{K}, \epsilon) \leq N_{Tn}(\mathcal{K}, \epsilon) \leq \Big(\frac{eT^2n^2B^2}{\epsilon d_\phi}\Big)^{d_\phi} \tag{A.10}$$

for any kernel family $\mathcal{K}$ bounded by B^2 with pseudodimension d_ϕ. Now we can prove Lemma 1:

Proof of Lemma 1. Fix some $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T) \in \mathcal{X}^{(T,n)}$. Define $\epsilon_K = \epsilon^2/4n$ and $\epsilon_{\mathcal{H}} = \epsilon/2$. Let $\widetilde{\mathcal{K}}$ be an ϵ_K-net of $\mathcal{K}$ with respect to $D_\infty^{\mathbf{x}}$. For every $\widetilde{K} \in \widetilde{\mathcal{K}}$ and every $t = 1 \ldots T$ let $\widetilde{\mathcal{H}}_{\widetilde{K}}^t$ be an $\epsilon_{\mathcal{H}}$-net of $\widetilde{\mathcal{H}}_{\widetilde{K}}$with respect to $d_\infty^{\mathbf{x}_t}$. Now fix some $\mathbf{h} \in \mathbb{H}^T$. Then there exists a kernel K such that $\mathbf{h} = (h_1, \ldots, h_T) \in \mathcal{H}_K^T$. Therefore there exists a kernel $\widetilde{K} \in \widetilde{\mathcal{K}}$ such that $|K_{\mathbf{x}_t} - \widetilde{K}_{\mathbf{x}_t}|_\infty < \epsilon_K$ for every t. By Lemma 1 in [89] $h_t(x_t) = K_{\mathbf{x}_t^t}^{1/2} w_t$ for some unit norm vector w_t for every t. Therefore for $\tilde{h}_t(\mathbf{x}_t)\widetilde{K}_{\mathbf{x}_t^t}^{1/2} w_t \in \mathcal{H}_{\widetilde{K}}$ we obtain that:

$$d_\infty^{\mathbf{x}_t}(h_t, \tilde{h}_t) = \max_i |f_t(x_i^t) - \tilde{h}_t(x_i^t)| \leq ||h_t(\mathbf{x}_t) - \tilde{h}_t(\mathbf{x}_t)|| =$$
$$||K_{\mathbf{x}_t}^{1/2} w_t - \widetilde{K}_{\mathbf{x}_t}^{1/2} w_t|| \leq \sqrt{n|K_{\mathbf{x}_t} - \widetilde{K}_{\mathbf{x}_t}|_\infty} \leq \sqrt{n\epsilon_K}.$$

In addition, for every $\tilde{h}_t \in \mathcal{H}_{\widetilde{K}}$ there exists $\tilde{\tilde{h}}_t \in \widetilde{\mathcal{H}}_{\widetilde{K}}^t$ such that $d_\infty^{\mathbf{x}_t}(\tilde{h}_t, \tilde{\tilde{h}}_t) < \epsilon_F$. Finally, if

we define $\tilde{\tilde{\mathbf{h}}} = (\tilde{\tilde{h}}_1, \ldots, \tilde{\tilde{h}}_T) \in \widetilde{\mathcal{H}}^1_{\widetilde{K}} \times \cdots \times \widetilde{\mathcal{H}}^T_{\widetilde{K}}$, we obtain:

$$d^{\mathbf{x}}_{\infty}(\mathbf{h}, \tilde{\tilde{\mathbf{f}}}) = \max_t d^{\mathbf{x}_t}_{\infty}(h_t, \tilde{\tilde{h}}_t) \leq \max_t (d^{\mathbf{x}_t}_{\infty}(h_t, \tilde{h}_t) + d^{\mathbf{x_t}}_{\infty}(\tilde{h}_t, \tilde{\tilde{h}}_t)) < \sqrt{n\epsilon_K} + \epsilon_F = \epsilon.$$

The above shows that $\widetilde{\mathcal{H}_{\mathcal{K}}} = \cup_{\widetilde{K} \in \widetilde{\mathcal{K}}} \widetilde{\mathcal{H}}^1_{\widetilde{K}} \times \cdots \times \widetilde{\mathcal{H}}^T_{\widetilde{K}}$ is an ϵ-net of $\mathbb{H}^T$ with respect to $\mathbf{x}$. Now the statement follows from (A.10) and the fact that for any $\mathcal{H}_K$ with bounded by B^2 kernel K([89, 4]):

$$N_n(\mathcal{H}_K, \epsilon) \leq 2\Big(\frac{4nB^2}{\epsilon^2}\Big)^{\frac{16B^2}{\epsilon^2}\log_2\left(\frac{\epsilon e n}{4B}\right)}. \tag{A.11}$$

□

Theorem 5 follows from combining Theorems 16, 17 and Lemma 1.

A.2 Proofs of Theorems 6 and 7

We start with analyzing the case of the sequential multi-task learning with the task order being fixed in advance. Without loss of generality we can consider the permutation π being the identity and obtain the following result:

Theorem 18. *For any deterministic transfer algorithm $\mathcal{TA}$, any deterministic learning algorithm $\mathcal{A}$, any prior distribution P and any $\delta > 0$, the following inequality holds with probability at least $1-\delta$ over the training sets $S_1, \ldots, S_T$ of size n each:*

$$\mathrm{er}_{\mathbf{D}}(\mathbf{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{Q}) + \frac{1}{T\sqrt{n}} \sum_{t=1}^{T} \mathrm{KL}(Q_t, ||P_t) + \frac{1}{8\sqrt{n}} + \frac{\log 1/\delta}{T\sqrt{n}}, \tag{A.12}$$

where

$$Q_t = \mathcal{A}(S_t, P_t) \tag{A.13}$$

$$P_t = \begin{cases} P \ \ \textit{for} \ \ t = 1 \\ \mathcal{TA}(S_1, \ldots, S_{t-1}) \ \ \textit{for} \ \ t \geq 2 \end{cases} . \tag{A.14}$$

Proof. We start with Donsker-Varadhan's variational formula [28] to change the expectation over posteriors $(Q_1, ..., Q_T)$ to the expectation over priors $(P_1, P_2, ..., P_T)$:

$$\begin{aligned} \mathrm{er}_{\mathbf{D}}(\mathbf{Q}) - \widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{Q}) \leq \ & \frac{1}{\lambda} \Big(\mathrm{KL}\left(Q_1 \times \cdots \times Q_T || P_1 \times \cdots \times P_T\right) \\ & + \log \mathop{\mathbb{E}}_{h_1 \sim P_1} \ldots \mathop{\mathbb{E}}_{h_T \sim P_T} \exp\left(\frac{\lambda}{T} \sum_{t=1}^{T} (\mathrm{er}_{D_t}(h_t) - \widehat{\mathrm{er}}_{S_t}(h_t)) \right) \Big) . \end{aligned} \tag{A.15}$$

This inequality holds for any $\lambda > 0$.

Note, that by construction P_t may depend on $S_1, ..., S_{t-1}$, but does not depend on $S_t, ..., S_T$. Therefore:

$$\begin{aligned} & \mathop{\mathbb{E}}_{S_1 \ldots S_T} \mathop{\mathbb{E}}_{h_1 \sim P_1} \ldots \mathop{\mathbb{E}}_{h_T \sim P_T} \exp\left(\frac{\lambda}{T} \sum_{t=1}^{T} (\mathrm{er}_{D_t}(h_t) - \widehat{\mathrm{er}}_{S_t}(h_t)) \right) = \\ & \mathop{\mathbb{E}}_{h_1 \sim P_1} \mathop{\mathbb{E}}_{S_1} \exp\left(\frac{\lambda}{T} (\mathrm{er}_{D_1}(h_1) - \widehat{\mathrm{er}}_{S_1}(h_1)) \right) \cdots \mathop{\mathbb{E}}_{h_T \sim P_T} \mathop{\mathbb{E}}_{S_T} \exp\left(\frac{\lambda}{T} (\mathrm{er}_{D_T}(h_T) - \widehat{\mathrm{er}}_{S_T}(h_T)) \right). \end{aligned} \tag{A.16}$$

Fix some $h_T \in \mathcal{H}$. Then we can rewrite the last term of (A.16) in the following way:

$$\exp\left(\frac{\lambda}{T}(\mathrm{er}_{D_T}(h_T) - \widehat{\mathrm{er}}_{S_T}(h_T))\right) = \prod_{t=1}^{T} \exp\left(\frac{\lambda}{Tn}\left(\mathrm{er}_{D_T}(h_T) - \ell(h_T(x_i^T), y_i^T)\right)\right). \quad \text{(A.17)}$$

Since the data points in S_T are i.i.d., all terms in this product are independent and take values between $\frac{\lambda(\mathrm{er}_{D_T}(h_T)-1)}{Tn}$ and $\frac{\lambda\,\mathrm{er}_{D_T}(h_T)}{Tn}$. Therefore, by Hoeffding's lemma [38], we obtain that the last term of (A.16) is bounded by a constant:

$$\mathop{\mathbb{E}}_{h_n\sim P_T}\mathop{\mathbb{E}}_{S_T} \exp\left(\frac{\lambda}{T}(\mathrm{er}_{D_T}(h_T) - \widehat{\mathrm{er}}_{S_T}(h_T))\right) \leq \exp\left(\frac{\lambda^2}{8T^2n}\right).$$

By repeating the same procedure for all other tasks we obtain that:

$$\mathop{\mathbb{E}}_{S_1\ldots S_T}\mathop{\mathbb{E}}_{h_1\sim P_1}\ldots\mathop{\mathbb{E}}_{h_T\sim P_T} \exp\left(\frac{\lambda}{T}\sum_{t=1}^{T}(er_{D_t}(h_t) - \widehat{\mathrm{er}}_{S_t}(h_t))\right) \leq \exp\left(\frac{\lambda^2}{8Tn}\right).$$

Therefore, by Markov's inequality, with probability at least $1-\delta$:

$$\mathop{\mathbb{E}}_{h_1\sim P_1}\ldots\mathop{\mathbb{E}}_{h_T\sim P_T} \exp\left(\frac{\lambda}{T}\sum_{t=1}^{T}(er_{D_t}(h_t) - \widehat{\mathrm{er}}_{S_t}(h_t))\right) \leq \frac{1}{\delta}\exp\left(\frac{\lambda^2}{8Tn}\right). \quad \text{(A.18)}$$

By combining (A.18) with (A.15) we get:

$$\mathrm{er}_{\mathbf{D}}(\mathbf{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathbf{Q}) + \frac{1}{\lambda}\,\mathrm{KL}\left(Q_1\times\cdots\times Q_T||P_1\times\cdots\times P_T\right) + \frac{\lambda}{8Tn} - \frac{1}{\lambda}\log\delta.$$

By setting $\lambda = T\sqrt{n}$ we obtain the final result. □

Note that Theorem 18 can be applied to any, fixed in advance order of tasks. Thus, if we apply it to every task order (of which there are $T!$ many) with confidence parameter $\delta/T!$ and combine the results using the union bound argument, we will obtain Theorem 6.

In exactly the same way one can prove an analog of Theorem 18 for the sequential multi-task learning with multiple subsequences for the case when the task order and the set of flags are fixed in advance. In order to obtain Theorem 7 one can use the same union bound argument, but with confidence parameter $\delta/((2T)^T)$, because there are more possible subsequences then sequences on T tasks, but there are not more than $2^{T-1}\cdot T! \leq (2T)^T$ of them.

A.3 Proof of Theorem 8

For any $I = \{t_1, \ldots, t_k\}$, any assignment $C = (c_1, \ldots, c_T)$ with $c_i \in I$ and any h_i by Proposition 1 the following inequality holds:

$$\frac{1}{T}\sum_{t=1}^{T} \mathrm{er}_{D_t}(h_{c_t}) \leq \frac{1}{T}\sum_{t=1}^{T} \mathrm{er}_{D_{c_t}}(h_{c_t}) + \frac{1}{T}\sum_{t=1}^{T} \mathrm{disc}(D_t, D_{c_t}) + \frac{1}{T}\sum_{t=1}^{T} \lambda_{tc_t}. \tag{A.19}$$

We continue with upper-bounding the expectations on the right-hand side of the above inequality by their empirical counterparts.

1. Bound $\frac{1}{T}\sum_{t=1}^{T} \mathrm{disc}(D_t, D_{c_t})$

We start with the discrepancy terms. Note that we only need to upper bound the discrepancies between pairs of labeled and unlabeled tasks. This is because the labeled tasks are assigned to themselves and thus the corresponding discrepancies are zero. Therefore there are only $T - k$ non-zero components. In order to control them we use Proposition 2 and a union bound argument. As a result we obtain that with probability at least $1 - \delta/2$ uniformly for all assignments C:

$$\frac{1}{T}\sum_{t=1}^{T} \mathrm{disc}(D_t, D_{c_t}) \leq \frac{1}{T}\sum_{t=1}^{T} \mathrm{disc}(S_t, S_{c_t}) + \frac{2(T-k)}{T}\sqrt{\frac{2d\log(2n) + \log(4T^2/\delta)}{n}}. \tag{A.20}$$

2. Bound $\frac{1}{T}\sum_{t=1}^{T} \mathrm{er}_{D_{c_t}}(h_{c_t})$

Next we bound the error term and do it in two steps. We introduce an intermediate quantity $\frac{1}{T}\sum_{t=1}^{T} \widehat{\mathrm{er}}_{S^u_{c_t}}(h_{c_t})$, where:

$$\widehat{\mathrm{er}}_{S^u_t}(h_t) = \frac{1}{n}\sum_{i=1}^{n} \ell(h_t(x_i^t), f_t(x_i^t), \tag{A.21}$$

which can be seen as a training error if the learner would receive labels for all n points for every of the selected tasks. The key observation is that the assignment C can depend only on the unlabeled data, while the hypotheses h_{c_t} can depend on the labels as well. The two-step procedure of going through the $\frac{1}{T}\sum_{t=1}^{T} \widehat{\mathrm{er}}_{S^u_{c_t}}(h_{c_t})$ allows us to isolate the

effects of these two sources of randomness.

2.1 Relate $\frac{1}{T}\sum_{t=1}^{T}\mathrm{er}_{D_{c_t}}(h_{c_t})$ to $\frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{er}}_{S^u_{c_t}}(h_{c_t})$

Because the choice of the assignment C and of the hypotheses h_t depends on the unlabeled data, we need a bound that holds simultaneously for all possible C and h_t. For that we apply Theorem 1 to each task and use a union bound argument. As a result we obtain that with probability at least $1-\delta/4$ for all assignments c and all hypotheses h_{c_t} the following holds:

$$\frac{1}{T}\sum_{t=1}^{T}\mathrm{er}_{D_{c_t}}(h_{c_t}) \leq \frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{er}}_{S^u_{c_t}}(h_{c_t}) + \sqrt{\frac{2d\log(en/d)}{n}} + \sqrt{\frac{\log(4T/\delta)}{2n}}. \tag{A.22}$$

2.2 Relate $\frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{er}}_{S^u_{c_t}}(h_{c_t})$ to $\frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{er}}_{S^l_{c_t}}(h_{c_t})$

Fix the unlabeled samples $S^u_1,\ldots,S^u_T$. This uniquely determines the chosen tasks I and the assignment $C=(c_1,\ldots,c_T)$, so the only remaining source of randomness is the uncertainty which subsets of the selected tasks are labeled. Analyzing this would be rather straightforward if the labeled points, S^l_i, were sampled i.i.d. from S^u_i (i.e. randomly *with replacement*). This is not the case, however, since we assume that exactly m points are labeled, i.e. S^l_i is sampled from S^u_i randomly *without replacement*, and this introduces dependencies between the elements.

For notational simplicity we pretend that exactly the first k tasks were selected, i.e. $I=\{1,\ldots,k\}$. The general case can be obtained by changing the indices in the proof from $1,\ldots,k$ to $i_1,\ldots,i_k$. For every $t=1,\ldots,k$ define p_t to be the number of times it appears in the assignment C.

To deal with the dependencies between the labeled data points we first note that any random labeled subset $S^l_i=(\bar{s}^i_1,\ldots,\bar{s}^i_m)$ can be described as the first m elements of a random permutation $Z_i=(z^i_1,\ldots,z^i_n)$ over n elements that correspond to the unlabeled sample S^u_i, i.e. $\bar{s}^i_j=(\bar{x}^i_j,\bar{y}^i_j)=(x^i_{z^i_j},y^i_{z^i_j})$. With this notation and writing $\mathbf{Z}=(Z_1,\ldots,Z_k)$

and $\ell(h, z_j^i) = \ell(h(\bar{x}_j^i), \bar{y}_j^i)$ we define the following function:

$$\Phi(\mathbf{Z}) = \sup_{h_1,\ldots,h_k} \frac{1}{T} \sum_{t=1}^{T} \left(\widehat{\mathrm{er}}_{S_{c_t}^u}(h_{c_t}) - \widehat{\mathrm{er}}_{S_{c_t}^l}(h_{c_t}) \right) = \tag{A.23}$$

$$\sup_{h_1,\ldots,h_k} \frac{1}{T} \sum_{i=1}^{k} p_i \left(\frac{1}{n} \sum_{j=1}^{n} \ell(h_t, z_j^i) - \frac{1}{m} \sum_{j=1}^{m} \ell(h_t, z_j^i) \right). \tag{A.24}$$

In order to establish a large deviation bound for Φ we use McDiarmid's inequality (Lemma 10) for martingales.

2.2.1 Construct a martingale sequence

For this, we interpret $\mathbf{Z} = (z_1^1, z_2^1, \ldots, z_n^k)$ as a sequence of kn dependent variables, $z_{11}, \ldots, z_{kn}$. For the sake of notational consistency we will keep using double indices, with the convention that the sample index, $j = 1, \ldots, n$, runs faster than the task index, $i = 1, \ldots, k$. Segments of a sequence will be denoted by upper and lower double indices, $z_{ij}^{\overline{ij}} = (z_{ij}, z_{i(j+1)}, \ldots, z_{\overline{ij}})$ for $ij \leq \overline{ij}$ and $z_{ij}^{\overline{ij}} = \emptyset$ otherwise. We now create a martingale sequence using Doob's construction [29]:

$$W_{ij} = \mathop{\mathbb{E}}_{\mathbf{Z}}\{\Phi(\mathbf{Z}) |\, z_{11}^{ij}\}. \tag{A.25}$$

where here and in the following when taking expectations over $\mathbf{Z}$ it is silently assumed that the expectation is taken only with respect to variables that are not conditioned on. Note that because of this convention, the expectations in (A.25) is only with respect to $z_{i(j+1)}, \ldots, z_{kn}$, so each W_{ij} is a random variable of $z_{11}, \ldots, z_{ij}$. In particular, $W_{00} = \mathbb{E}_{\mathbf{Z}} \Phi(\mathbf{Z})$ and $W_{kn} = \Phi(\mathbf{Z})$, and the in-between sequence is a martingale with respect to $z_{11}, \ldots, z_{kn}$:

$$\mathop{\mathbb{E}}_{\mathbf{Z}}\{\, W_{ij} | z_{11}^{i(j-1)} \} = \mathop{\mathbb{E}}_{\mathbf{Z}} \left\{ \mathop{\mathbb{E}}_{\mathbf{Z}}\{\Phi(\mathbf{Z}) |\, z_{11}^{ij}\} \middle|\, z_{11}^{i(j-1)} \right\} = \mathop{\mathbb{E}}_{\mathbf{Z}}\{\Phi(\mathbf{Z}) | z_{11}^{i(j-1)}\} = W_{i(j-1)}. \tag{A.26}$$

2.2.2. Compute an upper bound on the coefficient $\widehat{R}^2$

Let $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, n\}$ be fixed and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ be specific

permutations of n elements for which we use the same index conventions as for $\mathbf{Z}$. By σ and τ will denote elements in $\pi^{in}_{i(j+1)}$, i.e. σ and τ do not occur in any of the first j positions of the permutation π_i. Then

$$\begin{aligned} r_{ij}(\pi_{11}^{i(j-1)}) &= \sup_{\sigma \in \pi^{in}_{i(j+1)}} \{\, W_{ij} : z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)},\ z_{ij} = \sigma \} \\ &\quad - \inf_{\sigma \in \pi^{in}_{i(j+1)}} \{\, W_{ij} : z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)},\ z_{ij} = \sigma \} \\ = \sup_{\sigma \in \pi^{in}_{i(j+1)}} \sup_{\tau \in \pi^{in}_{i(j+1)}} &\Big[\mathop{\mathbb{E}}_{z^{kn}_{i(j+1)}} \{\Phi(\pi_{11}^{i(j-1)}, \sigma, z^{kn}_{i(j+1)})\} - \mathop{\mathbb{E}}_{z^{kn}_{i(j+1)}} \{\Phi(\pi_{11}^{i(j-1)}, \tau, z^{kn}_{i(j+1)})\} \Big]. \end{aligned} \tag{A.27}$$

To analyze (A.27) further, recall that:

$$\begin{aligned} &\mathop{\mathbb{E}}_{z^{kn}_{i(j+1)}} \{\Phi(\pi_{11}^{i(j-1)}, \sigma, z^{kn}_{i(j+1)})\} \\ = \sum_{\pi^{kn}_{i(j+1)}} &\Phi(\pi_{11}^{i(j-1)}, \sigma, \pi^{kn}_{i(j+1)}) \times \Pr(\ z^{kn}_{i(j+1)} = \pi^{kn}_{i(j+1)}\ | z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)} \wedge z_{ij} = \sigma\ ), \end{aligned}$$

where here and in the following we use the convention that sums over parts of $\boldsymbol{\pi}$ run only over values that lead to valid permutations. Because the permutations of different tasks are independent, this is equal to

$$\begin{aligned} = \sum_{\pi^{kn}_{i(j+1)}} &\Phi(\pi_{11}^{i(j-1)}, \sigma, \pi^{kn}_{i(j+1)}) \times \Pr(\ z^{in}_{i(j+1)} = \pi^{in}_{i(j+1)}\ | z_{i1}^{i(j-1)} = \pi_{i1}^{i(j-1)} \wedge z_{ij} = \sigma\ ) \\ &\times \Pr(z^{kn}_{(i+1)1} = \pi^{kn}_{(i+1)1}) \end{aligned} \tag{A.28}$$

We make the following observation: for any fixed π^{ij}_{i1} and any $\tau \notin \pi^{ij}_{i1}$, we can rephrase a summation over $\pi^{in}_{i(j+1)}$ into a sum over all positions where τ can occur, and a sum over all configuration for the entries that are not τ:

$$\sum_{\pi^{in}_{i(j+1)}} F(\pi^{in}_{i(j+1)}) = \sum_{l=j+1}^{n} \sum_{\pi^{i(l-1)}_{i(j+1)}} \sum_{\pi^{in}_{i(l+1)}} F(\pi^{i(l-1)}_{i(j+1)}, \tau, \pi^{in}_{i(l+1)}) \tag{A.29}$$

for any function F. Applying this to the summation in (A.28), we obtain

$$\begin{aligned}
&\sum_{\pi_{i(j+1)}^{kn}} \Phi(\pi_{11}^{i(j-1)}, \sigma, \pi_{i(j+1)}^{kn}) \Pr(\, z_{i(j+1)}^{in} = \pi_{i(j+1)}^{in} \,|z_{i1}^{i(j-1)} = \pi_{i1}^{i(j-1)} \wedge z_{ij} = \sigma \,) \\
&\quad \times \Pr(z_{(i+1)1}^{kn} = \pi_{(i+1)1}^{kn}) = \sum_{l=j+1}^{n} \sum_{\pi_{i(j+1)}^{i(l-1)}} \sum_{\pi_{i(l+1)}^{kn}} \Phi(\pi_{11}^{i(j-1)}, \sigma, \pi_{i(j+1)}^{i(l-1)}, \tau, \pi_{i(l+1)}^{kn}) \\
&\quad \times \Pr(\, z_{i(j+1)}^{i(l-1)} = \pi_{i(j+1)}^{i(l-1)} \wedge z_{i(l+1)}^{kn} = \pi_{i(l+1)}^{kn} | z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)} \wedge z_{ij} = \sigma \wedge z_{il} = \tau) \\
&\quad \times \Pr(z_{(i+1)1}^{kn} = \pi_{(i+1)1}^{kn}) = \mathop{\mathbb{E}}_{l \sim U_{j+1}^{n}} \mathop{\mathbb{E}}_{\mathbf{Z}} \Phi(\mathbf{Z} | z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)} \wedge z_{ij} = \sigma \wedge z_{il} = \tau), \qquad \text{(A.30)}
\end{aligned}$$

where U_{j+1}^{n} denotes the uniform distribution over the set $\{j+1, \ldots, n\}$. The analogue derivation can be applied to the quantity in line (A.27) with σ and τ exchanged.

For any $\mathbf{Z}$ denote by $\mathbf{Z}^{ij \leftrightarrow il}$ the permutation obtained by switching z_{ij} and z_{il}. Then, due to the linearity of the expectation:

$$r_{ij}(\pi_{11}^{i(j-1)}) = \sup_{\sigma,\tau} \mathop{\mathbb{E}}_{l \sim U_{j+1}^{n}} \mathop{\mathbb{E}}_{\mathbf{Z}} \Big[\Phi(\mathbf{Z}) - \Phi(\mathbf{Z}^{ij \leftrightarrow il}) | z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)}, z_{ij} = \sigma, z_{il} = \tau) \Big]. \qquad \text{(A.31)}$$

From the definition of Φ we see that $\Phi(\mathbf{Z}) - \Phi(\mathbf{Z}^{ij \leftrightarrow il}) = 0$ when $j, l \in \{1, \ldots, m\}$ or $j, l \in \{m+1 \ldots, n\}$. Since $l > j$ in (A.31) this implies $r_{ij}(\pi_{11}^{i(j-1)}) = 0$ for $j \in \{m+1, \ldots, n\}$. The only remaining cases are $j \in \{1, \ldots, m\}$ and $l \in \{m+1, \ldots, n\}$, for which we obtain

$$\Phi(\mathbf{Z}) - \Phi(\mathbf{Z}^{ij \leftrightarrow il}) \leq \sup_{h_1, \ldots, h_k} \frac{1}{T} \cdot p_i \cdot \frac{1}{m} (-\ell(h_i, z_j^i) + \ell(h_i, z_l^i)) \leq \frac{p_i}{Tm}, \qquad \text{(A.32)}$$

where for the first inequality we used that $\sup F - \sup G \leq \sup(F - G)$ for any F, G, and for the second inequality we used that ℓ is bounded by $[0, 1]$. Consequently, $r_{ij}(\pi_{11}^{i(j-1)}) \leq \frac{n-m}{n-j} \frac{p_i}{Tm}$ in this case. Therefore

$$\begin{aligned}
\widehat{R}^2 &= \sum_{i=1}^{k} \sum_{j=1}^{n} \left(r_{ij}(\pi_{11}^{i(j-1)}) \right)^2 \leq \frac{1}{T^2 m^2} \sum_{j=1}^{m} \left(\frac{n-m}{n-j} \right)^2 \sum_{i=1}^{k} p_i^2 \\
&= \frac{(n-m)^2}{T^2 m (n-0.5)(n-m-0.5)} \sum_{i=1}^{k} p_i^2.
\end{aligned}$$

Now from Lemma 10 we obtain that with probability at least $1 - \delta/4$:

$$\Phi(\mathbf{Z}) - \mathop{\mathbb{E}}_{\mathbf{Z}} \Phi(\mathbf{Z}) = W_{kn} - W_0 \leq \frac{1}{T}\sqrt{\sum_{i=1}^{k} p_i^2}\sqrt{\frac{\log(4/\delta)}{2m}} \cdot \frac{n-m}{\sqrt{(n-0.5)(n-m-0.5)}}. \quad \text{(A.33)}$$

2.2.3 Bound $\mathbb{E}_{\mathbf{Z}}\, \Phi(\mathbf{Z})$

The main ingredient here is Lemma 11. First we rewrite $\Phi(\mathbf{Z})$ in the following way:

$$\Phi(\mathbf{Z}) = \frac{1}{T}\sum_{i=1}^{k} \sup_h p_i(\widehat{\mathrm{er}}_{S_i^u}(h) - \widehat{\mathrm{er}}_{S_i^l}(h)) = \frac{1}{Tm}\sum_{i=1}^{k} \Phi_i(Z),$$
$$\Phi_i(\mathbf{Z}) = \sup_h m\, p_i(\widehat{\mathrm{er}}_{S_i^u}(h) - \widehat{\mathrm{er}}_{S_i^l}(h)).$$

Note that even though $\mathcal{H}$ can be infinitely large, we can identify a finite subset that represents all possible predictions of hypotheses in $\mathcal{H}$ on $S_1^u \cup \cdots \cup S_k^u$. We denote their number by $L \leq 2^{kn}$ and the corresponding hypotheses by $h^1, \ldots, h^L$.

Let $i \in \{1, \ldots, k\}$ be fixed. Define a set of n L-dimensional vectors, $V_i = \{v_{i1}, \ldots, v_{in}\}$, where for every $j \in \{1, \ldots, n\}$:

$$v_{ij} = \Big[\, p_i\big(\widehat{\mathrm{er}}_{S_i^u}(h^1) - \ell(h^1(x_j^i), y_j^i)\big),\ \ldots,\ p_i\big(\widehat{\mathrm{er}}_{S_i^u}(h^L) - \ell(h^L(x_j^i), y_j^i)\big)\Big]. \quad \text{(A.34)}$$

With this notation, choosing a random subset $S_i^l \subset S_i^u$ corresponds to sampling m vectors from V_i uniformly without replacement.

Let $U_i = \{u_{i1}, \ldots, u_{im}\}$ be sampled from V_i in that way. Then

$$\Phi_i(Z) = F\left(\sum_{j=1}^{m} u_{ij}\right), \quad \text{(A.35)}$$

where the function F takes as input an L-dimensional vector and returns the value of its maximum component. We now bound $\mathbb{E}_Z\, \Phi_i(Z)$ by applying Lemma 11, because by Lemma 12 $F(x)$ is a convex function:

$$\mathop{\mathbb{E}}_{Z} \Phi_t(Z) = \mathop{\mathbb{E}}_{U_i}\ F\Big(\sum_{j=1}^{m} u_{ij}\Big) \leq \mathop{\mathbb{E}}_{\hat{U}_i}\left[F\left(\sum_{j=1}^{m} \hat{u}_{ij}\right)\right]. \quad \text{(A.36)}$$

Switching from the U_i sets by the $\hat{U}_i$ sets in Φ corresponds to switching from random subsets S_i^l to random sets $\tilde{S}_i$ consisting of m points sampled from S_i^l uniformly *with* replacement. Therefore we obtain

$$\mathop{\mathbb{E}}_{Z} \Phi_i(Z) = \mathop{\mathbb{E}}_{S_i^l} \Phi_i(S_i^l) \leq \mathop{\mathbb{E}}_{\tilde{S}_i} \Phi_i(\tilde{S}_i), \tag{A.37}$$

which allows us to continue analyzing $\mathbb{E}_Z \Phi_i(Z)$ in the standard way using Rademacher complexities and independent samples. Applying the common symmetrization trick and introducing Rademacher random variables σ_j we obtain:

$$\Phi_i(\tilde{S}_i) \leq 2 \mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{j=1}^{m} \sigma_j p_i \ell(h(x_j^i), y_j^i).$$

We can rewrite this using the fact that $\ell(y, y') = [\![y \neq y']\!] = \frac{1-yy'}{2}$

$$\mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{j=1}^{m} \sigma_j p_i \ell(h(x_j^i), y_j^i) = \mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{j=1}^{m} \sigma_j p_i \frac{1 - h(x_j^i) y_j^i}{2} = \frac{1}{2} \mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{j=1}^{m} -\sigma_j y_j^i p_i h(x_j^i).$$

Since $-\sigma_j y_j^i$ has the same distribution as σ_j:

$$= \frac{p_i}{2} \mathop{\mathbb{E}}_{\sigma} \sup_{h \in A} \sum_{j=1}^{m} \sigma_j h(x_j^i),$$

where $A = \{(h(x_1^i), \ldots, h(x_m^i)) : \ h \in \mathcal{H}\}$. According to Sauer's lemma (Corollary 3.3 in [66]):

$$|A| \leq \left(\frac{em}{d}\right)^d. \tag{A.38}$$

At the same time:

$$\|a\|_2 = \sqrt{\sum_{j=1}^{m} (h(x_j^i))^2} = \sqrt{m}. \tag{A.39}$$

Therefore, by Massart's lemma (Theorem 3.3 in [66]):

$$\mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{j=1}^{m} \sigma_j p_i \ell(h(x_j^i), y_j^i) \leq \frac{p_i}{2} \sqrt{2dm \log(em/d)}. \tag{A.40}$$

By applying this result for all i we obtain:

$$\mathop{\mathbb{E}}_{\mathbf{Z}} \Phi(\mathbf{Z}) = \frac{1}{Tm} \sum_{i=1}^{k} \mathop{\mathbb{E}}_{\mathbf{Z}} \Phi_i(\mathbf{Z}) \leq \frac{1}{Tm} \sum_{i=1}^{k} \mathop{\mathbb{E}}_{\tilde{S}_i} \Phi_i(\tilde{S}_i) \leq \frac{1}{T} \sum_{i=1}^{k} p_i \sqrt{\frac{2d \log(em/d)}{m}} = \sqrt{\frac{2d \log(em/d)}{m}}. \tag{A.41}$$

Therefore we obtain that for fixed unlabeled samples $S_1^u, \ldots, S_T^u$ with probability at least $1 - \delta/4$ for all choices of $h_{c_1}, \ldots, h_{c_T}$:

$$\frac{1}{T} \sum_{t=1}^{T} \widehat{\mathrm{er}}_{S_{c_t}^u}(h_{c_t}) \leq \frac{1}{T} \sum_{t=1}^{T} \widehat{\mathrm{er}}_{S_{c_t}^l}(h_{c_t}) + \sqrt{\frac{2d \log(em/d)}{m}} + \frac{1}{T} \sqrt{\sum_{i=1}^{k} p_i^2} \sqrt{\frac{\log(4/\delta)}{2m}} \cdot \frac{n-m}{\sqrt{(n-0.5)(n-m-0.5)}}. \tag{A.42}$$

Theorem 8 follows by combining inequalities (A.19), (A.20), (A.22) and (A.42).

A.4 Proof of Theorem 9

As in the proof of Theorem 8, we bound the multi-task error by the errors on the source tasks, and transition to empirical quantities while keeping the effect of random sampling controlled. However, the steps will be more involved, since we now require the bounds to be uniform also in the (continuous) weights α.

1. Obtain an analog of (A.19)

We start with establishing an analog of (A.19) for the case of multi-source transfer.

Fix a subset of labeled tasks $I = \{i_1, \ldots, i_k\}$, a task $\langle D_t, f_t \rangle$ and a weight vector $\alpha \in \Lambda^I$. Let $h_i^* \in \arg\min_{h \in \mathcal{H}} (\mathrm{er}_t(h) + \mathrm{er}_i(h))$.[1] Writing $\ell(h, h')$ as shorthand for $\ell(h(x), h'(x))$,

[1] If the minimum is not attained, the same inequality follows by an argument of arbitrary close approximation.

we have

$$|\operatorname{er}_\alpha(h) - \operatorname{er}_t(h)| = \Big|\sum_{i\in I}\alpha_i \operatorname{er}_i(h) - \operatorname{er}_t(h)\Big| \le \sum_{i\in I}\alpha_i\big|\operatorname{er}_i(h) - \operatorname{er}_t(h)\big| \tag{A.43}$$

$$\le \sum_{i\in I}\alpha_i\Big(\big|\operatorname{er}_i(h) - \mathop{\mathbb{E}}_{x\sim D_i}\ell(h,h_i^*)\big| + \big|\mathop{\mathbb{E}}_{x\sim D_i}\ell(h,h_i^*) - \mathop{\mathbb{E}}_{x\sim D_t}\ell(h,h_i^*)\big| \tag{A.44}$$

$$+\big|\operatorname{er}_t(h) - \mathop{\mathbb{E}}_{x\sim D_t}\ell(h,h_i^*)\big|\Big) = (*) \tag{A.45}$$

We can bound each summand:

$$|\operatorname{er}_i(h) - \mathop{\mathbb{E}}_{x\sim D_i}\ell(h,h_i^*)| \le \operatorname{er}_i(h^*) \text{ by the triangular inequality for } \ell$$

$$|\mathop{\mathbb{E}}_{x\sim D_i}\ell(h,h_i^*) - \mathop{\mathbb{E}}_{x\sim D_t}\ell(h,h_i^*)\big| \le \operatorname{disc}(D_i,D_t) \text{ by the definition of discrepancy}$$

$$|\operatorname{er}_t(h) - \mathop{\mathbb{E}}_{x\sim D_t}\ell(h,h_i^*)\big| \le \operatorname{er}_t(h_i^*) \text{ by the triangular inequality for } \ell$$

Therefore,

$$(*) \le \sum_{i\in I}\alpha_i(\operatorname{er}_i(h_i^*) + \operatorname{disc}(D_i,D_t) + \operatorname{er}_t(h_i^*)) = \sum_{i\in I}\alpha_i(\lambda_{it} + \operatorname{disc}(D_i,D_t)). \tag{A.46}$$

Consequently, assuming that every task t has its own weights α^t we obtain that:

$$\frac{1}{T}\sum_{t=1}^T \operatorname{er}_t(h) \le \frac{1}{T}\sum_{t=1}^T \operatorname{er}_{\alpha^t}(h_t) + \frac{1}{T}\sum_{t=1}^T\sum_{i\in I}\alpha_i^t \operatorname{disc}(D_t,D_i) + \frac{1}{T}\sum_{t=1}^T\sum_{i\in I}\alpha_i^t\lambda_{ti}. \tag{A.47}$$

We continue with bounding every expectation on the right hand side of (A.47) by its empirical counterpart.

2. Bound $\frac{1}{T}\sum_{t=1}^T\sum_{i\in I}\alpha_i^t \operatorname{disc}(D_t,D_i)$

As in the proof of Theorem 8, we apply Proposition 2 to every summand and combine the results using a union bound argument. We obtain that with probability at least $1-\delta/2$ uniformly for all choices of I and $\alpha^1,\ldots,\alpha^T \in \Lambda^I$:

$$\frac{1}{T}\sum_{t=1}^T\sum_{i\in I}\alpha_i^t \operatorname{disc}(D_t,D_i) \le \frac{1}{T}\sum_{t=1}^T\sum_{i\in I}\alpha_i^t \operatorname{disc}(S_t,S_i) + 2\sqrt{\frac{2d\log(2n)+\log(4T^2/\delta)}{n}}. \tag{A.48}$$

3. Bound $\frac{1}{T}\sum_{t=1}^{T}\mathrm{er}_{\alpha^t}(h_t)$

Now we upper-bound the error term in two steps, analogously to the proof of Theorem 8.

3.1 Relate $\frac{1}{T}\sum_{t=1}^{T}\mathrm{er}_{\alpha^t}(h_t)$ to $\frac{1}{T}\sum_{t=1}^{T}\tilde{\mathrm{er}}_{\alpha^t}(h_t)$

We start with relating the multi-task error to the hypothetical empirical error, if the learner would receive labels for all examples in the selected labeled tasks:

$$\tilde{\mathrm{er}}_{\alpha}(h)=\sum_{i\in I}\alpha_i\widehat{\mathrm{er}}_{S_i^u}(h)\quad\text{for}\quad\widehat{\mathrm{er}}_{S_i^u}(h)=\frac{1}{n}\sum_{j=1}^{n}\ell(h(x_j^i),f_i(x_j^i)). \tag{A.49}$$

Clearly, if $m=n$ this part is not necessary and we can avoid the resulting complexity terms.

Because the choice of the tasks to label, I, their weights, $\boldsymbol{\alpha}=(\alpha^1,\ldots,\alpha^T)$, and the predictors, $\mathbf{h}=(h_1,\ldots,h_T)$, all depend on the unlabeled data, we aim for a bound that is holds simultaneous for all choices of these quantities, under the condition that I and $\boldsymbol{\alpha}$ depend only on the unlabeled samples, while $\mathbf{h}$ can be chosen based also on the labeled subsets.

Our main tool is a refined version of McDiarmid's inequality, due to Maurer [56] (Lemma 13), which allows us to make use of the internal structure of the weights, $\boldsymbol{\alpha}$, while deriving a large deviation bound.

For any $\mathbf{S}=(S_1^u,\ldots,S_T^u)$ define:

$$\Psi(\mathbf{S})=\sup_{I=\{i_1,\ldots,i_k\}}\sup_{\alpha^1,\ldots,\alpha^n\in\Lambda^I}\sup_{h_1,\ldots,h_T}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{T}\alpha_i^t(\mathrm{er}_i(h_t)-\widehat{\mathrm{er}}_{S_i^u}(h_t)) \tag{A.50}$$

$$=\sup_{I}\sup_{\boldsymbol{\alpha}}\sup_{\mathbf{h}}\ g(\boldsymbol{\alpha},\mathbf{h},\mathbf{S}) \tag{A.51}$$

for

$$g(\boldsymbol{\alpha},\mathbf{h},\mathbf{S})=\sum_{i=1}^{T}\sum_{j=1}^{n}\left(\frac{1}{Tn}\sum_{t=1}^{T}\alpha_i^t(\mathrm{er}_i(h_t)-\ell(h_t(x_j^i),f_t(x_j^i)))\right). \tag{A.52}$$

For notational simplicity we will sometimes think of every S_t^u as a set of *pairs* (x_i^t, y_i^t), where $y_i^t = f_t(x_i^t)$. To apply Lemma 13 we establish a bound on $\Delta_{+,\Psi}(\mathbf{S}) = \sum_i \sum_j (\Psi(\mathbf{S}) - \Psi_{ij}(\mathbf{S}))^2$, with

$$\Psi_{ij}(\mathbf{S}) = \inf_{(x,y)} \sup_{\boldsymbol{\alpha}} \sup_{\mathbf{h}} \; g(\boldsymbol{\alpha}, \mathbf{h}, \mathbf{S} \setminus \{(x_j^i, y_j^i)\} \cup \{(x,y)\}, \tag{A.53}$$

i.e. the possible smallest value for Ψ when changing only the data point (x_j^i, y_j^i). Let $\boldsymbol{\alpha}^*, \mathbf{h}^*$ be the point where the $\sup$ in the (A.51) is attained[1], i.e. $\Psi(\mathbf{S}) = g(\boldsymbol{\alpha}^*, \mathbf{h}^*, \mathbf{S})$. Then:

$$\Psi_{ij}(\mathbf{S}) \geq \inf_{(x,y)} g(\boldsymbol{\alpha}^*, \mathbf{h}^*, \mathbf{S} \setminus \{(x_j^i, y_j^i)\} \cup \{(x,y)\} \;) \tag{A.54}$$

and therefore

$$\Psi(\mathbf{S}) - \Psi_{ij}(\mathbf{S}) \leq g(\boldsymbol{\alpha}^*, \mathbf{h}^*, \mathbf{S}) - \inf_{(x,y)} g(\boldsymbol{\alpha}^*, \mathbf{h}^*, \mathbf{S} \setminus \{(x_j^i, y_j^i)\} \cup \{(x,y)\}) \tag{A.55}$$

$$\leq \sup_{(x,y)} \frac{1}{Tn} \sum_{t=1}^{T} \alpha_i^{*t}(-\ell(h_t^*(x_j^i), y_j^i) + \ell(h_t^*(x), y)) \;\leq\; \frac{1}{Tn} \sum_{t=1}^{T} \alpha_i^{*t}, \tag{A.56}$$

where for the last inequality we use that ℓ is bounded in $[0,1]$. Because also $\Psi(\mathbf{S}) - \Psi_{ij}(\mathbf{S}) \geq 0$, we obtain

$$\Delta_{+,\Psi}(\mathbf{S}) = \sum_{i=1}^{T} \sum_{j=1}^{n} (\Psi(\mathbf{S}) - \Psi_{ij}(\mathbf{S}))^2 \leq \sum_{i=1}^{T} \sum_{j=1}^{n} \frac{1}{T^2 n^2} \left(\sum_{t=1}^{T} \alpha_i^{*t} \right)^2 \tag{A.57}$$

$$\leq \frac{1}{T^2 n} \left(\sum_{i=1}^{T} \sum_{t=1}^{T} \alpha_i^{*t} \right)^2 = \frac{1}{n}, \tag{A.58}$$

(remember that $\sum_i \alpha_i = 1$ for any $\alpha \in \Lambda^I$). Therefore, according to Lemma 13 with probability at least $1 - \delta/4$:

$$\Psi(\mathbf{S}) \leq \mathbb{E}\; \Psi(\mathbf{S}) + \sqrt{\frac{2}{n} \log \frac{4}{\delta}}. \tag{A.59}$$

[1] If the supremum is not attained the subsequent inequality still follows from an argument of arbitrarily close approximation.

To bound $\mathbb{E}_S \Psi(\mathbf{S})$ we use symmetrization and Rademacher variables, σ_{ij}:

$$\mathop{\mathbb{E}}_{\mathbf{S}} \Psi(\mathbf{S}) = \mathop{\mathbb{E}}_{\mathbf{S}} \sup_I \sup_{\alpha^1,\ldots,\alpha^T \in \Lambda^I} \sup_{h_1,\ldots,h_T} \sum_{i=1}^{T}\sum_{j=1}^{n} \left(\frac{1}{Tn} \sum_{t=1}^{T} \alpha_i^t (\mathrm{er}_i(h_t) - \ell(h_t(x_j^i), y_j^i)) \right) \tag{A.60}$$

$$\leq 2 \mathop{\mathbb{E}}_{\mathbf{S}} \mathop{\mathbb{E}}_{\sigma} \sup_I \sup_{\alpha^1,\ldots,\alpha^T \in \Lambda^I} \sup_{h_1,\ldots,h_T} \sum_{i=1}^{T}\sum_{j=1}^{n} \left(\frac{\sigma_{ij}}{Tn} \sum_{t=1}^{T} \alpha_i^t \ell(h_t(x_j^i), y_j^i) \right) \tag{A.61}$$

$$\leq 2 \mathop{\mathbb{E}}_{\mathbf{S}} \mathop{\mathbb{E}}_{\sigma} \frac{1}{T} \sum_{t=1}^{T} \sup_{\alpha^t \in \Lambda, h_t} \sum_{i=1}^{T}\sum_{j=1}^{n} \frac{\sigma_{ij}\alpha_i^t}{n} \sum_{t=1}^{T} \ell(h_t(x_j^i), y_j^i) \tag{A.62}$$

$$\leq 2 \mathop{\mathbb{E}}_{\mathbf{S}} \mathop{\mathbb{E}}_{\sigma} \sup_{\alpha,h} \sum_{i=1}^{T}\sum_{j=1}^{n} \frac{\sigma_{ij}\alpha_i}{n} \ell(h(x_j^i), y^i), \tag{A.63}$$

where line (A.62) is obtained from line (A.61) by dropping the assumption of a common sparsity pattern between the α-s. Note that the function inside the last $\sup$ is linear in $\alpha \in \Lambda$, therefore $\sup_\alpha$ can be reduced to the $\sup$ over the corners of the simplex, $\{(1,0,\ldots,0), \ldots, (0,\ldots,0,1)\}$. At the same time, by Sauer's lemma, the number of different choices of h on $\mathbf{S}$ is bounded by $\left(\frac{eTn}{d}\right)^d$. Therefore, the total number of different choices in (A.63) is bounded by $T\left(\frac{enT}{d}\right)^d$. Furthermore, for any choice of α and h, the norm of the Tn-vector formed by the summands of (A.63) is bounded by $1/\sqrt{n}$, because

$$\sum_{i=1}^{T}\sum_{j=1}^{n} \left(\frac{\sigma_{ij}\alpha_i}{n} \ell(h(x_j^i), y^i) \right)^2 = \frac{1}{n^2} \sum_{i=1}^{T}\sum_{j=1}^{n} \left(\alpha_i \ell(h(x_j^i), y^i) \right)^2 \tag{A.64}$$

$$\leq \frac{1}{n^2} \sum_{j=1}^{n} \left(\sum_{i=1}^{T} \alpha_i \right)^2 = \frac{1}{n}. \tag{A.65}$$

Therefore, by Massart's lemma:

$$\mathop{\mathbb{E}}_{\sigma} \sup_{\alpha,h} \sum_{i=1}^{T}\sum_{j=1}^{n} \frac{\sigma_{il}\alpha_i}{n} \ell(h(x_l^i), y_l^i) \leq \frac{\sqrt{2(\log T + d\log(enT/d))}}{\sqrt{n}}. \tag{A.66}$$

Combining (A.59) and (A.66) we obtain that with probability at least $1 - \delta/4$ simultaneously for all choices of tasks to be labeled, I, weights $\boldsymbol{\alpha}$ and hypotheses $\mathbf{h}$:

$$\frac{1}{T}\sum_{t=1}^{T} \mathrm{er}_{\alpha^t}(h_t) \leq \frac{1}{T}\sum_{t=1}^{T} \tilde{\mathrm{er}}_{\alpha^t}(h_t) + \sqrt{\frac{8(\log T + d\log(enT/d))}{n}} + \sqrt{\frac{2}{n}\log\frac{4}{\delta}}. \tag{A.67}$$

3.2 Relate $\frac{1}{T}\sum_{t=1}^{T}\widehat{\mathrm{er}}_{\alpha^t}(h_t)$ to $\frac{1}{T}\sum_{t=1}^{T}\tilde{\mathrm{er}}_{\alpha^t}(h_t)$

Fix the unlabeled samples $S_1^u,\ldots,S_T^u$. This uniquely determines the chosen tasks I and the weights $\alpha^1,\ldots,\alpha^T\in\Lambda^I$, so the only remaining source of randomness is the uncertainty which subsets of the selected tasks are labeled.

For notational simplicity we pretend that exactly the first k tasks were selected, i.e. $I=\{1,\ldots,k\}$. The general case can be obtained by changing the indices in the proof from $1,\ldots,k$ to $i_1,\ldots,i_k$.

To deal with the dependencies between the labeled data points we first note that any random labeled subset $S_i^l=(\bar{s}_1^i,\ldots,\bar{s}_m^i)$ can be described as the first m elements of a random permutation $Z_i=(z_1^i,\ldots,z_n^i)$ over n elements that correspond to the unlabeled sample S_i^u, i.e. $\bar{s}_j^i=(\bar{x}_j^i,\bar{y}_j^i)=(x^i_{z_j^i},y^i_{z_j^i})$. With this notation and writing $\mathbf{Z}=(Z_1,\ldots,Z_k)$ and $\ell(h,z_j^i)=\ell(h(\bar{x}_j^i),\bar{y}_j^i)$ we define the following function

$$\Phi(\mathbf{Z})=\sup_{h_1,\ldots,h_T}\frac{1}{T}\sum_{t=1}^{T}\tilde{\mathrm{er}}_{\alpha^t}(h_t)-\widehat{\mathrm{er}}_{\alpha^t}(h_t) \tag{A.68}$$

$$=\sup_{h_1,\ldots,h_T}\sum_{i=1}^{k}\frac{1}{T}\sum_{t=1}^{T}\alpha_i^t\Big(\frac{1}{n}\sum_{j=1}^{n}\ell(h_t,z_j^i)-\frac{1}{m}\sum_{j=1}^{m}\ell(h_t,z_j^i)\Big). \tag{A.69}$$

Our main tool is McDiarmid's inequality (Lemma 10) for martingales.

3.2.1 Construct a martingale sequence

For this, we interpret $\mathbf{Z}=(z_1^1,z_2^1,\ldots,z_n^k)$ as a sequence of kn dependent variables, $z_{11},\ldots,z_{kn}$. For the sake of notational consistency we will keep using double indices, with the convention that the sample index, $j=1,\ldots,n$, runs faster than the task index, $i=1,\ldots,k$. Segments of a sequence will be denoted by upper and lower double indices, $z_{ij}^{\overline{ij}}=(z_{ij},z_{i(j+1)},\ldots,z_{\overline{ij}})$ for $ij\le\overline{ij}$ and $z_{ij}^{\overline{ij}}=\emptyset$ otherwise. We now create a martingale sequence using Doob's construction [29]:

$$W_{ij}=\mathop{\mathbb{E}}_{\mathbf{Z}}\{\Phi(\mathbf{Z})|\ z_{11}^{ij}\ \}. \tag{A.70}$$

where here and in the following when taking expectations over $\mathbf{Z}$ it is silently assumed

that the expectation is taken only with respect to variables that are not conditioned on. Note that because of this convention, the expectations in (A.70) is only with respect to $z_{i(j+1)}, \ldots, z_{kn}$, so each W_{ij} is a random variable of $z_{11}, \ldots, z_{ij}$. In particular, $W_{00} = \mathbb{E}_{\mathbf{Z}} \Phi(\mathbf{Z})$ and $W_{kn} = \Phi(\mathbf{Z})$, and the in between sequence is a martingale with respect to $z_{11}, \ldots, z_{kn}$:

$$\mathop{\mathbb{E}}_{\mathbf{Z}}\{\ W_{ij}|z_{11}^{i(j-1)}\ \} = \mathop{\mathbb{E}}_{\mathbf{Z}}\{\ \mathop{\mathbb{E}}_{\mathbf{Z}}\{\Phi(\mathbf{Z})|\ z_{11}^{ij}\}\big|\ z_{11}^{i(j-1)}\ \} = \mathop{\mathbb{E}}_{\mathbf{Z}}\{\Phi(\mathbf{Z})|z_{11}^{i(j-1)}\ \} = W_{i(j-1)}. \qquad \text{(A.71)}$$

3.2.2 Upper-bound $\widehat{R}^2$

In order to apply Lemma 10 we need an upper bound on the coefficient $\widehat{R}^2$ defined there.

Let $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, n\}$ be fixed and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ be specific permutations of n elements for which we use the same index conventions as for $\mathbf{Z}$. By σ and τ will denote elements in $\pi_{i(j+1)}^{in}$, i.e. σ and τ do not occur in any of the first j positions of the permutation π_i. Then

$$\begin{aligned} r_{ij}(\pi_{11}^{i(j-1)}) &= \sup_{\sigma \in \pi_{i(j+1)}^{in}} \{\ W_{ij} : z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)},\ z_{ij} = \sigma\} \\ &\quad - \inf_{\sigma \in \pi_{i(j+1)}^{in}} \{\ W_{ij} : z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)},\ z_{ij} = \sigma\} \\ = \sup_{\sigma \in \pi_{i(j+1)}^{in}} \sup_{\tau \in \pi_{i(j+1)}^{in}} \Big[\mathop{\mathbb{E}}_{z_{i(j+1)}^{kn}} &\{\Phi(\pi_{11}^{i(j-1)}, \sigma, z_{i(j+1)}^{kn})\} - \mathop{\mathbb{E}}_{z_{i(j+1)}^{kn}} \{\Phi(\pi_{11}^{i(j-1)}, \tau, z_{i(j+1)}^{kn})\}\Big]. \end{aligned} \qquad \text{(A.72)}$$

To analyze (A.72) further, recall that:

$$\begin{aligned} &\mathop{\mathbb{E}}_{z_{i(j+1)}^{kn}} \{\Phi(\pi_{11}^{i(j-1)}, \sigma, z_{i(j+1)}^{kn})\} \\ &= \sum_{\pi_{i(j+1)}^{kn}} \Phi(\pi_{11}^{i(j-1)}, \sigma, \pi_{i(j+1)}^{kn}) \times \Pr(\ z_{i(j+1)}^{kn} = \pi_{i(j+1)}^{kn}\ |z_{11}^{i(j-1)} = \pi_{11}^{i(j-1)} \wedge z_{ij} = \sigma\ ), \end{aligned}$$

where here and in the following we use the convention that sums over parts of $\boldsymbol{\pi}$ run only over values that lead to valid permutations. Because the permutations of different

task are independent, this is equal to

$$= \sum_{\pi^{kn}_{i(j+1)}} \Phi(\pi^{i(j-1)}_{11}, \sigma, \pi^{kn}_{i(j+1)}) \tag{A.73}$$
$$\times \Pr(\, z^{in}_{i(j+1)} = \pi^{in}_{i(j+1)} \,|z^{i(j-1)}_{i1} = \pi^{i(j-1)}_{i1} \wedge z_{ij} = \sigma \,) \Pr(z^{kn}_{(i+1)1} = \pi^{kn}_{(i+1)1})$$

We make the following observation: for any fixed π^{ij}_{i1} and any $\tau \notin \pi^{ij}_{i1}$, we can rephrase a summation over $\pi^{in}_{i(j+1)}$ into a sum over all positions where τ can occur, and a sum over all configuration for the entries that are not τ:

$$\sum_{\pi^{in}_{i(j+1)}} F(\pi^{in}_{i(j+1)}) = \sum_{l=j+1}^{n} \sum_{\pi^{i(l-1)}_{i(j+1)}} \sum_{\pi^{in}_{i(l+1)}} F(\pi^{i(l-1)}_{i(j+1)}, \tau, \pi^{in}_{i(l+1)}) \tag{A.74}$$

for any function F. Applying this to the summation in (A.73), we obtain

$$\sum_{\pi^{kn}_{i(j+1)}} \Phi(\pi^{i(j-1)}_{11}, \sigma, \pi^{kn}_{i(j+1)}) \Pr(\, z^{in}_{i(j+1)} = \pi^{in}_{i(j+1)} \,|z^{i(j-1)}_{i1} = \pi^{i(j-1)}_{i1} \wedge z_{ij} = \sigma \,)$$
$$\times \Pr(z^{kn}_{(i+1)1} = \pi^{kn}_{(i+1)1}) = \sum_{l=j+1}^{n} \sum_{\pi^{i(l-1)}_{i(j+1)}} \sum_{\pi^{kn}_{i(l+1)}} \Phi(\pi^{i(j-1)}_{11}, \sigma, \pi^{i(l-1)}_{i(j+1)}, \tau, \pi^{kn}_{i(l+1)})$$

$$\times \Pr(\, z^{i(l-1)}_{i(j+1)} = \pi^{i(l-1)}_{i(j+1)} \wedge z^{kn}_{i(l+1)} = \pi^{kn}_{i(l+1)} | z^{i(j-1)}_{11} = \pi^{i(j-1)}_{11} \wedge z_{ij} = \sigma \wedge z_{il} = \tau)$$
$$\times \Pr(z^{kn}_{(i+1)1} = \pi^{kn}_{(i+1)1}) = \mathop{\mathbb{E}}_{l \sim U^n_{j+1}} \mathop{\mathbb{E}}_{\mathbf{Z}} \Phi(\mathbf{Z} | z^{i(j-1)}_{11} = \pi^{i(j-1)}_{11} \wedge z_{ij} = \sigma \wedge z_{il} = \tau),$$

where U^n_{j+1} denotes the uniform distribution over the set $\{j+1, \ldots, n\}$. The analogue derivation can be applied to the quantity in line (A.72) with σ and τ exchanged.

For any $\mathbf{Z}$ denote by $\mathbf{Z}^{ij \leftrightarrow il}$ the permutation obtained by switching z_{ij} and z_{il}. Then, due to the linearity of the expectation:

$$r_{ij}(\pi^{i(j-1)}_{11}) = \sup_{\sigma,\tau} \{ \mathop{\mathbb{E}}_{l \sim U^n_{j+1}} \mathop{\mathbb{E}}_{\mathbf{Z}} \{ \Phi(\mathbf{Z}) - \Phi(\mathbf{Z}^{ij \leftrightarrow il}) | z^{i(j-1)}_{11} = \pi^{i(j-1)}_{11}, z_{ij} = \sigma, z_{il} = \tau). \tag{A.75}$$

From the definition of Φ we see that $\Phi(\mathbf{Z}) - \Phi(\mathbf{Z}^{ij \leftrightarrow il}) = 0$ when $j, l \in \{1, \ldots, m\}$ or $j, l \in \{m+1 \ldots, n\}$. Since $l > j$ in (A.75) this implies $r_{ij}(\pi^{i(j-1)}_{11}) = 0$ for $j \in \{m+1, \ldots, n\}$.

The only remaining cases are $j \in \{1, \dots, m\}$ and $l \in \{m+1, \dots, n\}$, for which we obtain

$$\Phi(\mathbf{Z}) - \Phi(\mathbf{Z}^{ij \leftrightarrow il}) \leq \sup_{h_1, \dots, h_T} \frac{1}{T} \sum_{t=1}^{T} \alpha_i^t \frac{1}{m} (-\ell(h_t, z_j^i) + \ell(h_t, z_l^i)) \leq \frac{1}{Tm} \sum_{t=1}^{T} \alpha_i^t.$$

where for the first inequality we used that $\sup F - \sup G \leq \sup(F - G)$ for any F, G, and for the second inequality we used that ℓ is bounded by $[0, 1]$. Consequently, $r_{ij}(\pi_{11}^{i(j-1)}) \leq \frac{n-m}{n-j} \frac{1}{Tm} \sum_{t=1}^{T} \alpha_i^t$ in this case. Therefore[1]

$$\widehat{R}^2 = \sum_{i=1}^{k} \sum_{j=1}^{n} \left(r_{ij}(\pi_{11}^{i(j-1)}) \right)^2 \leq \frac{1}{T^2 m^2} \sum_{j=1}^{m} \left(\frac{n-m}{n-j} \right)^2 \sum_{i=1}^{k} \left(\sum_{t=1}^{T} \alpha_i^t \right)^2 \tag{A.76}$$

$$\leq \frac{1}{T^2 m} \sum_{i=1}^{k} \left(\sum_{t=1}^{T} \alpha_i^t \right)^2. \tag{A.77}$$

3.2.3 Upper-bound $\mathbb{E}_{\mathbf{Z}} \Phi(\mathbf{Z})$

The main tool here is Lemma 11. First we rewrite $\Phi(\mathbf{Z})$ in the following way:

$$\Phi(\mathbf{Z}) = \frac{1}{T} \sum_{t=1}^{T} \sup_h \sum_{i=1}^{k} \alpha_i^t (\widehat{\mathrm{er}}_{S_i^u}(h) - \widehat{\mathrm{er}}_{S_i^l}(h)) = \frac{1}{Tm} \sum_{t=1}^{T} \Phi_t(\mathbf{Z})$$

$$\Phi_t(\mathbf{Z}) = \sup_h \sum_{i=1}^{k} m \, \alpha_i^t (\widehat{\mathrm{er}}_{S_i^u}(h) - \widehat{\mathrm{er}}_{S_i^l}(h)).$$

Note that even though $\mathcal{H}$ can be infinitely large, we can identify a finite subset that represents all possible predictions of hypothesis in $\mathcal{H}$ on $S_1^u \cup \dots \cup S_k^u$. We denote their number by $L \leq 2^{kn}$ and the corresponding hypotheses by $h^1, \dots, h^L$.

Let $t \in \{1, \dots, T\}$ be fixed. For every $i \in \{1, \dots, k\}$ define a set of n L-dimensional vectors, $V_i^t = \{v_{i1}^t, \dots, v_{in}^t\}$, where for every $j \in \{1, \dots, n\}$:

$$v_{ij}^t = \Big[\alpha_i^t \big(\tilde{\mathrm{er}}_i(h^1) - \ell(h^1(x_j^i), y_j^i)\big), \; \dots, \; \alpha_i^t \big(\tilde{\mathrm{er}}_i(h^L) - \ell(h^L(x_j^i), y_j^i)\big) \Big]. \tag{A.78}$$

With this notation, for every $i \in \{1, \dots, k\}$ choosing a random subset $S_i^l \subset S_i^u$ corresponds to sampling m vectors from V_i^t uniformly without replacement.

[1] We generously bound $\frac{n-m}{n-j} \leq 1$ in this step. By keeping the corresponding factor in the analysis one obtains that the constant B in the theorem can be improved at least by a factor of $\frac{(n-m)^2}{(n-0.5)(n-m-0.5)}$.

For every $i \in \{1, \dots, k\}$, let $U_i = \{u_{i1}, \dots, u_{im}\}$ be sampled from V_i^t in that way. Then

$$\Phi_t(\mathbf{Z}) = F\left(\sum_{i=1}^{k}\sum_{j=1}^{m} u_{ij}\right), \tag{A.79}$$

where the function F takes as input an L-dimensional vector and returns the value of its maximum component. We now bound $\mathbb{E}_Z \Phi_t(\mathbf{Z})$ by applying Lemma 11 k times:

$$\mathop{\mathbb{E}}_{Z} \Phi_t(\mathbf{Z}) = \mathop{\mathbb{E}}_{U_1,\dots,U_k} F\Big(\sum_{i=1}^{k}\sum_{j=1}^{m} u_{ij}\Big) \tag{A.80}$$

$$= \mathop{\mathbb{E}}_{U_1,\dots,U_{k-1}} \left[\mathop{\mathbb{E}}_{U_k}\left[F\Big(\sum_{i=1}^{k-1}\sum_{j=1}^{m} u_{ij} + \sum_{j=1}^{m} u_{kj}\Big)\Big| U_1, \dots, U_{k-1}\right]\right] \tag{A.81}$$

By Lemma 12 $F(x)$ is a convex function. Thus $F(\textit{const} + x)$ is also convex and we can apply Lemma 11 with respect to U_k.

$$\leq \mathop{\mathbb{E}}_{U_1,\dots,U_{k-1}} \left[\mathop{\mathbb{E}}_{\hat{U}_k}\left[F\left(\sum_{i=1}^{k-1}\sum_{j=1}^{m} u_{ij} + \sum_{j=1}^{m} \hat{u}_{kj}\right)\Big| U_1, \dots, U_{k-1}\right]\right] \tag{A.82}$$

where $\hat{U}_k = \{u_{ki}, \dots, u_{km}\}$ is a set of m vectors sampled from V_k^t *with replacement*.

$$= \mathop{\mathbb{E}}_{U_1,\dots,U_{k-1},\hat{U}_k} \left[F\left(\sum_{i=1}^{k-1}\sum_{j=1}^{m} u_{ij} + \sum_{j=1}^{m} \hat{u}_{kj}\right)\right]. \tag{A.83}$$

Repeating the process k times, we obtain

$$\leq \dots \leq \mathop{\mathbb{E}}_{\hat{U}_1,\dots,\hat{U}_k} \left[F\left(\sum_{i=1}^{k}\sum_{j=1}^{m} \hat{u}_{ij}\right)\right]. \tag{A.84}$$

Note that writing the conditioning in the above expressions is just for clarity of presentation, since the $U_1, \dots, U_k$ are actually independent of each other.

Switching from the U sets by the $\hat{U}$ sets in Φ corresponds to switching from random subsets S_i^l to random sets $\tilde{S}_i$ consisting of m points sampled from S_i^u uniformly *with* replacement. Therefore we obtain

$$\mathop{\mathbb{E}}_{Z} \Phi_t(Z) = \mathop{\mathbb{E}}_{S_1^l,\dots,S_k^l} \Phi_t(S_1^l, \dots, S_k^l) \leq \mathop{\mathbb{E}}_{\tilde{S}_1,\dots,\tilde{S}_k} \Phi_t(\tilde{S}_1, \dots, \tilde{S}_k), \tag{A.85}$$

which allows us to continue analyzing $\mathbb{E}_Z \Phi_t(\mathbf{Z})$ in the standard way using Rademacher complexities and independent samples. Applying the common symmetrization trick and introducing Rademacher random variables σ_{ij} we obtain

$$\Phi_t(\tilde{S}_1, \ldots, \tilde{S}_k) \leq 2 \mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{i=1}^{k} \sum_{j=1}^{m} \sigma_{ij} \alpha_i^t \ell(h(x_j^i), y_j^i).$$

We can rewrite this using the fact that $\ell(y, y') = [\![y \neq y']\!] = \frac{1-yy'}{2}$:

$$\begin{aligned} \mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{i=1}^{k} \sum_{j=1}^{m} \sigma_{ij} \alpha_i^t \ell(h(x_j^i), y_j^i) &= \mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{i=1}^{k} \sum_{j=1}^{m} \sigma_{ij} \alpha_i^t \frac{1 - h(x_j^i) y_j^i}{2} \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{i=1}^{k} \sum_{j=1}^{m} -\sigma_{ij} y_j^i \alpha_i^t h(x_j^i) \end{aligned}$$

Since $-\sigma_{ij} y_j^i$ has the same distribution as σ_{ij}:

$$= \frac{1}{2} \mathop{\mathbb{E}}_{\sigma} \sup_{a(h) \in A} \sum_{i=1}^{k} \sum_{j=1}^{m} \sigma_{ij} a_{ij}(h),$$

where $a_{ij}(h) = \alpha_i^t h(x_j^i)$ and $A = \{a(h) : \ h \in \mathcal{H}\}$. According to Sauer's lemma (Corollary 3.3 in [66]):

$$|A| \leq \left(\frac{ekm}{d} \right)^d. \tag{A.86}$$

At the same time:

$$\|a\|_2 = \sqrt{\sum_{i=1}^{k} \sum_{j=1}^{m} (\alpha_i^t h(x_j^i))^2} = \sqrt{m} \sqrt{\sum_{i=1}^{k} (\alpha_i^t)^2}. \tag{A.87}$$

Therefore, by Massart's lemma (Theorem 3.3 in [66]):

$$\mathop{\mathbb{E}}_{\sigma} \sup_h \sum_{i=1}^{k} \sum_{j=1}^{m} \sigma_{ij} \alpha_i^t \ell(h(x_j^i), y_j^i) \leq \frac{1}{2} \sqrt{\sum_{i=1}^{k} (\alpha_i^t)^2} \cdot \sqrt{2dm \log(ekm/d)}. \tag{A.88}$$

By applying this result for all t we obtain:

$$\mathop{\mathbb{E}}_{\mathbf{Z}} \Phi(\mathbf{Z}) = \frac{1}{Tm} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{Z}} \Phi_t(\mathbf{Z}) \leq \frac{1}{Tm} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\tilde{S}} \Phi_t(\tilde{S}) \tag{A.89}$$

$$\leq \frac{1}{T} \sum_{t=1}^{T} \sqrt{\sum_{i=1}^{k} (\alpha_i^t)^2} \cdot \sqrt{\frac{2d \log(ekm/d)}{m}}. \tag{A.90}$$

Combining (A.77) and (A.90) with Lemma 10 we obtain that for fixed unlabeled samples $S_1^u, \ldots, S_T^u$ with probability at least $1 - \delta/4$ for all choices of $h_1, \ldots, h_T$:

$$\frac{1}{T} \sum_{t=1}^{T} \tilde{\mathrm{er}}_{\alpha^t}(h_t) \leq \frac{1}{T} \sum_{t=1}^{T} \widehat{\mathrm{er}}_{\alpha^t}(h_t) + \frac{1}{T} \|\alpha\|_{2,1} \sqrt{\frac{2d \log(ekm/d)}{m}} + \frac{1}{T} \|\alpha\|_{1,2} \sqrt{\frac{\log(4/\delta)}{2m}}.$$

By further combining it with (A.67) we obtain that the following inequality holds uniformly in $h_1, \ldots, h_T \in \mathcal{H}$ with probability at least $1 - \delta/2$ over the sampling of the unlabeled training sets, $S_1^u, \ldots, S_T^u$, and labeled training sets, $(S_i^l)_{i \in I}$, provided that the subset of labeled tasks, $I \subset \{1, \ldots, T\}$, and the task weights, $\alpha^1, \ldots, \alpha^T \in \Lambda^I$, depend deterministically on the unlabeled training only.

$$\frac{1}{T} \sum_{t=1}^{T} \mathrm{er}_{\alpha^t}(h_t) \leq \frac{1}{T} \sum_{t=1}^{T} \widehat{\mathrm{er}}_{\alpha^t}(h_t) + \frac{1}{T} \|\alpha\|_{2,1} \sqrt{\frac{2d \log(ekm/d)}{m}} + \frac{1}{T} \|\alpha\|_{1,2} \sqrt{\frac{\log(4/\delta)}{2m}}$$

$$+ \sqrt{\frac{8(\log T + d \log(enT/d))}{n}} + \sqrt{\frac{2}{n} \log \frac{4}{\delta}}. \tag{A.91}$$

The statement of Theorem 9 follows by combining (A.47) with (A.48) and (A.91).

B Proofs of theorems in Chapter 4

B.1 Proof of Theorem 10

As in the proof of Theorem 5 we will use the technique of covering numbers to obtain Theorem 10. However, in this case we will need covers of the kernel family $\mathcal{K}$ with respect to a probability distribution. In particular, for any probability distribution D over $\mathcal{X} \times \mathcal{Y}$, we denote its projection on $\mathcal{X}$ by D_X and define the following distance between the kernels:

$$D_D(K, \tilde{K}) = \max\{\max_{h \in \mathcal{H}_K} \min_{h' \in \mathcal{H}_{\tilde{K}}} \mathop{\mathbb{E}}_{x \sim D_X} |h(x) - h'(x)|, \max_{h' \in \mathcal{H}_{\tilde{K}}} \min_{h \in \mathcal{H}_K} \mathop{\mathbb{E}}_{x \sim D_X} |h(x) - h'(x)|\}. \quad \text{(B.1)}$$

Similarly, for any set of T distributions $\mathbf{D} = (D_1, \ldots, D_T)$ we define:

$$D_{\mathbf{D}}(K, \tilde{K}) = \max_{t=1 \ldots T} D_{D_t}(K, \tilde{K}). \quad \text{(B.2)}$$

The minimal size of the corresponding ϵ-cover of a set of kernels $\mathcal{K}$ we will denote by $N_{D_{\mathbf{D}}}(\mathcal{K}, \epsilon)$ and the corresponding uniform covering number by by $N_{(D,T)}(\mathcal{K}, \epsilon) = \max_{(D_1, \ldots, D_T)} N_{D_{\mathbf{D}}}(\mathcal{K}, \epsilon)$.

Now we proceed similarly to the way Theorem 10 is proved by first obtaining a generalization bound in terms of the defined above covering numbers.

First, note that for a set of T data distributions $\mathbf{D} = (D_1, \ldots, D_T)$ the following inequality holds true:

$$\Pr\left\{\mathbf{S} \in (\mathcal{X} \times \mathcal{Y})^{(T,n)} \; \exists \, K \in \mathcal{K} : \mathrm{er}_{\mathbf{D}}(\mathcal{H}_K) > \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathcal{H}_K) + \epsilon\right\} \leq$$
$$\Pr\left\{\mathbf{S} \in (\mathcal{X} \times \mathcal{Y})^{(T,n)} \; \exists \, \mathbf{h} \in \mathbb{H}^T : \mathrm{er}_{\mathbf{D}}(\mathbf{h}) > \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathbf{h}) + \epsilon\right\},$$

where

$$\mathrm{er}_{\mathbf{D}}(\mathcal{H}_K) = \frac{1}{T} \sum_{t=1}^{T} \inf_{h \in \mathcal{H}_K} \mathop{\mathbb{E}}_{(x,y) \sim D_t} [\![yh(x) < 0]\!]. \quad \text{(B.3)}$$

Thus, by exactly following the proof of Theorem 16 one can obtain that:

$$\Pr\left\{\exists K \in \mathcal{K} \;\; \mathrm{er}^{\gamma/2}_{\mathbf{D}}(\mathcal{H}_K) - \widehat{\mathrm{er}}^{\gamma}_{\mathbf{S}}(\mathcal{H}_K) > \frac{\epsilon}{2}\right\} < 2 N_{(T,2n)}(\mathbb{H}^T, \gamma/4) \exp\left(-\frac{Tn\epsilon^2}{32}\right). \quad \text{(B.4)}$$

Therefore the only thing that is left is a bound on the difference between $\mathrm{er}_{\mathfrak{D}}(\mathcal{H}_K)$ and $\mathrm{er}^{\gamma}_{\mathbf{D}}(\mathcal{H}_K)$.

We will use the following notation:

$$\mathrm{er}_D(\mathcal{H}_K) = \inf_{h\in\mathcal{H}_K} \mathop{\mathbb{E}}_{(x,y)\sim D} [\![h(x)y < 0]\!],$$
$$\mathrm{er}^\gamma_D(\mathcal{H}_K) = \inf_{h\in\mathcal{H}_K} \mathop{\mathbb{E}}_{(x,y)\sim D} [\![h(x)y < \gamma]\!]$$

and proceed in a way analogous to the proof of Theorem 16. First, if we define:

$$Q = \{\mathbf{D} = (D_1, \ldots, D_T) : \ \exists \mathcal{H}_K : \ \ \mathrm{er}_{\mathfrak{D}}(\mathcal{H}_K) > \mathrm{er}^\gamma_{\mathbf{D}}(\mathcal{H}_K) + \epsilon\}$$
$$R = \{\mathbf{D} = (\mathbf{D_1}, \mathbf{D_2}) : \ \exists \mathcal{H}_K : \ \ \mathrm{er}_{\mathbf{D_2}}(\mathcal{H}_K) > \mathrm{er}^\gamma_{\mathbf{D_1}}(\mathcal{H}_K) + \epsilon/2\},$$

then according to the symmetrization argument $\Pr(Q) \le 2\Pr(R)$.

Now, if we define Γ_{2T} to be a set of permutations σ on a set $\{1, 2, \ldots, 2T\}$, such that $\{\sigma(t), \sigma(T+t)\} = \{t, T+t\}$ for all $t = 1 \ldots T$, we obtain that $\Pr(R) \le \max_{\mathbf{D}} \Pr_\sigma(\sigma\mathbf{D} \in R)$, if $T > 2/\epsilon^2$. So, the only thing that is left is a reduction to a finite class.

Fix $\mathbf{D} = (\mathbf{D_1}, \mathbf{D_2}) = (D_1, \ldots, D_{2T})$ and denote by $\tilde{\mathcal{K}} \subset \mathcal{K}$ a set of kernels, such that for every $K \in \mathcal{K}$ there exists a $\tilde{K} \in \tilde{\mathcal{K}}$ such that:

$$\mathrm{er}^\gamma_{D_t}(\mathcal{H}_K) + \epsilon/8 \ge \mathrm{er}^{\gamma/2}_{D_t}(\mathcal{H}_{\tilde{K}}) \ge \mathrm{er}_{D_t}(\mathcal{H}_K) - \epsilon/8 \ \ \forall t = 1 \ldots 2T. \tag{B.5}$$

Then, if $\mathcal{F}_K$ is such that

$$\mathrm{er}_{\mathbf{D_2}}(\mathcal{H}_K) > \mathrm{er}^\gamma_{\mathbf{D_1}}(\mathcal{H}_K) + \epsilon/2,$$

then the corresponding $\tilde{K}$ satisfies

$$\mathrm{er}^{\gamma/2}_{\mathbf{D_2}}(\mathcal{H}_{\tilde{K}}) > \mathrm{er}^{\gamma/2}_{\mathbf{D_1}}(\mathcal{H}_{\tilde{K}}) + \epsilon/4.$$

Therefore:

$$\Pr_\sigma\{\sigma\mathbf{D} \in R\} \le \Pr_\sigma\left\{\exists K \in \tilde{\mathcal{K}} : \frac{1}{T}\sum_{t=1}^{T}(\mathrm{er}^{\gamma/2}_{D_{\sigma(T+t)}}(\mathcal{H}_K) - \mathrm{er}^{\gamma/2}_{D_{\sigma(t)}}(\mathcal{H}_K)) > \epsilon/4\right\} \le$$
$$|\tilde{\mathcal{K}}| \max_{K\in\tilde{\mathcal{K}}} \Pr_\sigma\left\{\frac{1}{T}\sum_{t=1}^{T}(\mathrm{er}^{\gamma/2}_{D_{\sigma(T+t)}}(\mathcal{H}_K) - \mathrm{er}^{\gamma/2}_{D_{\sigma(t)}}(\mathcal{H}_K)) > \epsilon/4\right\} =$$
$$|\tilde{\mathcal{K}}| \max_{K\in\tilde{\mathcal{K}}} \Pr_\beta\left\{\frac{1}{T}\sum_{t=1}^{T}|\,\mathrm{er}^{\gamma/2}_{D_{T+t}}(\mathcal{H}_K) - \mathrm{er}^{\gamma/2}_{D_t}(\mathcal{H}_K)|\beta_t > \epsilon/4\right\} = (*),$$

where β_t are independent random variables uniformly distributed over $\{-1, +1\}$. Therefore $\{|\operatorname{er}^{\gamma/2}_{D_{T+t}}(\mathcal{H}_K) - \operatorname{er}^{\gamma/2}_{D_t}(\mathcal{H}_K)|\beta_t\}$ are T independent random variables that take values between -1 and 1 and have zero mean. Therefore by applying Hoeffding's inequality we obtain:

$$(*) \leq |\tilde{\mathcal{K}}| \exp\left(-\frac{2T^2\epsilon^2/16}{4T}\right) = |\tilde{\mathcal{K}}| \exp\left(-\frac{T\epsilon^2}{32}\right). \tag{B.6}$$

Now we show that $|\tilde{\mathcal{K}}|$ can be upper bounded using covering number $N_{(D,2T)}$:

Lemma 2. *For any set of probability distributions $\mathbf{D} = (P_1, \ldots, P_{2T})$ there exists $\tilde{\mathcal{K}}$ that satisfies condition of equation B.5 and $|\tilde{\mathcal{K}}| \leq N_{(D,2T)}(\mathcal{K}, \epsilon\gamma/16)$.*

Proof. Fix a set of distributions $\mathbf{D} = (D_1, \ldots, P_{2T})$ and denote by $\tilde{\mathcal{K}}$ an $\epsilon\gamma/16$-cover of $\mathcal{K}$ with respect to $D_{\mathbf{D}}$. Then $|\tilde{\mathcal{K}}| \leq N_{(D,2T)}(\mathcal{K}, \epsilon\gamma/16)$. By definition of a cover for any kernel $K \in \mathcal{K}$ there exists $\tilde{K} \in \tilde{\mathcal{K}}$ such that $D_{\mathbf{D}}(K, \tilde{K}) < \epsilon\gamma/16$. Equivalently, it means that for every $K \in \mathcal{K}$ there exists $\tilde{K} \in \tilde{\mathcal{K}}$ such that the following two conditions hold for every $t = 1, \ldots, 2T$:

$$1.\forall\ h \in \mathcal{H}_K\ \ \exists h' \in \mathcal{H}_{\tilde{K}}: \quad \mathop{\mathbb{E}}_{(x,y)\sim D_t}(|h(x) - h'(x)|) < \frac{\epsilon\gamma}{16}, \tag{B.7}$$

$$2.\forall\ h' \in \mathcal{H}_{\tilde{K}}\ \ \exists h \in \mathcal{H}_K: \quad \mathop{\mathbb{E}}_{(x,y)\sim D_t}(|h(x) - h'(x)|) < \frac{\epsilon\gamma}{16}. \tag{B.8}$$

Fix some K and the corresponding kernel $\tilde{K}$ from the cover and take any D_t. By Markov's inequality applied to the first condition we obtain that for every $h \in \mathcal{H}_K$ there exists a $h' \in \mathcal{H}_{\tilde{K}}$ such that $\Pr\{(x, y) \sim D_t:\ |h(x) - h'(x)| > \gamma/2\} < \epsilon/8$. Then $\operatorname{er}^{\gamma/2}_{D_t}(h') \leq \operatorname{er}^{\gamma}_{D_t}(h) + \epsilon/8$. By applying the same argument to the second condition we conclude that for every $h' \in \mathcal{H}_{\tilde{K}}$ there exists a $h \in \mathcal{H}_K$ such that $\Pr\{(x, y) \sim D_t:\ |h(x) - h'(x)| > \gamma/2\} < \epsilon/8$. Then $\operatorname{er}_{D_t}(h) \leq \operatorname{er}^{\gamma/2}_{D_t}(h') + \epsilon/8$. By definition of infinum in $\operatorname{er}^{2\gamma}_{D_t}(\mathcal{H}_K)$ for every δ there exists $h \in \mathcal{H}_K$ such that $\operatorname{er}^{\gamma}_{D_t}(\mathcal{H}_K) + \delta > \operatorname{er}^{\gamma}_{D_t}(h) \geq \operatorname{er}^{\gamma}_{D_t}(\mathcal{H}_K)$. By above construction for such h there exists $h' \in \mathcal{H}_{\tilde{K}}$ such that $\operatorname{er}^{\gamma}_{D_t}(h) \geq \operatorname{er}^{\gamma/2}_{D_t}(h') - \epsilon/8 \geq \operatorname{er}^{\gamma/2}_{D_t}(\mathcal{H}_{\tilde{K}}) - \epsilon/8$. By combining these inequalities we obtain that for every $\delta > 0$:

$$\operatorname{er}^{\gamma}_{D_t}(\mathcal{H}_K) + \delta > \operatorname{er}^{\gamma/2}_{D_t}(\mathcal{H}_{\tilde{K}}) - \epsilon/8,$$

or, equivalently:

$$\operatorname{er}^{\gamma}_{D_t}(\mathcal{H}_K) \geq \operatorname{er}^{\gamma/2}_{D_t}(\mathcal{H}_{\tilde{K}}) - \epsilon/8.$$

Analogously we can get that

$$\mathrm{er}_{D_t}^{\gamma/2}(\mathcal{H}_{\tilde{K}}) \geq \mathrm{er}_{D_t}(\mathcal{H}_K) - \epsilon/8.$$

So, we obtain condition (B.5). □

By combining the above Lemma with (B.6) we obtain the following result (the second inequality can be obtained in a similar manner):

Theorem 19. *For any $\epsilon > 0$, if $T > 2/\epsilon^2$, the following holds:*

$$\Pr\left\{\exists K \in \mathcal{K}: \ \ \mathrm{er}_{\mathfrak{D}}(\mathcal{H}_K) > \mathrm{er}_{\mathbf{D}}^{\gamma}(\mathcal{H}_K) + \epsilon\right\} \leq 2N_{(D,2T)}(\mathcal{K}, \epsilon\gamma/16)\exp\left(-\frac{T\epsilon^2}{32}\right),$$
$$\Pr\left\{\exists K \in \mathcal{K}: \mathrm{er}_{\mathfrak{D}}^{2\gamma}(\mathcal{H}_K) < \mathrm{er}_{\mathbf{D}}^{\gamma}(\mathcal{H}_K) - \epsilon\right\} \leq 2N_{(D,2T)}(\mathcal{K}, \epsilon\gamma/16)\exp\left(-\frac{T\epsilon^2}{32}\right).$$

Now we need to bound covering numbers $N_{(D,T)}$ in terms of the pseudodimension of the kernel family $\mathcal{K}$.

Lemma 3. *There exists a constant C such that for any kernel family $\mathcal{K}$ with pseudodimension d_ϕ such that $K(x,x) \leq B^2$ for every $K \in \mathcal{K}$ and every $x \in \mathcal{X}$:*

$$N_{(D,T)}(\mathcal{K}, \epsilon) \leq \left(CT^5 d_\phi^5 \Big(B/\epsilon\Big)^{17}\right)^{d_\phi}. \tag{B.9}$$

The proof this result is based on the following lemma that connects sample-based and distribution-based covers of kernel families:

Lemma 4. *For any probability distribution D over $\mathcal{X} \times \mathcal{Y}$ and any B^2-bounded set of kernels $\mathcal{K}$ with pseudo-dimension d_ϕ there exists a sample $\mathbf{x}$ of size $n = cd_\phi^2 B^5/\epsilon^5$ for some constant c, such that for every $K, \tilde{K}$ if $D_1^{\mathbf{x}}(K, \tilde{K}) < \epsilon/2$, then $D_D(K, \tilde{K}) < \epsilon$ (where $D_1^{\mathbf{x}}$ is the same as D_D, but all expectations over D are substituted by empirical averages over $\mathbf{x}$).*

Proof. Define $G = \left\{ g : \mathcal{X} \to [0,1] : g(x) = \frac{|h(x)-h'(x)|}{B} \text{ for some } h, h' \in \cup_{K \in \mathcal{K}} \mathcal{H}_K \right\}$. Then (using Lemma 2 and 3 in [12] and Theorem 1 in [89]):

$$\Pr\left\{ \mathbf{x} \in \mathcal{X}^n : \ \exists K, \tilde{K} : |D_1^{\mathbf{x}}(K, \tilde{K}) - D_D(K, \tilde{K})| > \epsilon/2 \right\} \leq$$

$$\Pr\left\{ \mathbf{x} \in \mathcal{X}^n \text{: } \exists h, h' \in \cup \mathcal{H}_K : \left| \frac{1}{n} \sum_{i=1}^{n} |h(x_i) - h'(x_i)| - \mathop{\mathbb{E}}_{(x,y)\sim D} |h(x) - h'(x)| \right| > \frac{\epsilon}{2} \right\} =$$

$$\Pr\left\{ \mathbf{x} \in \mathcal{X}^n : \ \ \exists g, g' \in G : \left| \frac{1}{n} \sum_{i=1}^{n} g(x_i) - \mathop{\mathbb{E}}_{(x,y)\sim D} g(x) \right| > \epsilon/2B \right\} \leq$$

$$4 \max_{\mathbf{x}} N\left(\frac{\epsilon/32}{B}, G, d_1^{\mathbf{x}} \right) e^{-\epsilon^2 n/512B^2} \leq 4 \max_{\mathbf{x}} N\left(\frac{\epsilon}{64B}, \cup \mathcal{H}_K / B, d_1^{\mathbf{x}} \right)^2 e^{-\epsilon^2 n/512B^2} =$$

$$4 \max_{\mathbf{x}} N(\epsilon/64, \cup \mathcal{H}_K, d_1^{\mathbf{x}})^2 e^{-\epsilon^2 n/512B^2} \leq 4 \max_{\mathbf{x}} N(\epsilon/64, \cup \mathcal{H}_K, d_\infty^{\mathbf{x}})^2 e^{-\epsilon^2 n/512B^2} \leq$$

$$4 \cdot 4 \cdot N(\mathcal{K}, \epsilon^2/(64^2 \cdot 4n))^2 \cdot \left(\frac{16nB^2 64^2}{\epsilon^2} \right)^{\frac{2 \cdot 64^3 B^2}{\epsilon^2} \log\left(\frac{\epsilon e n}{64 * 8B}\right)} e^{-\epsilon^2 n/512B^2} \leq$$

$$16 \left(\frac{2^{14} e n^3 B^2}{\epsilon^2 d_\phi} \right)^{2d_\phi} \left(\frac{2^{16} n B^2}{\epsilon^2} \right)^{\frac{2^{19} B^2}{\epsilon^2} \log\left(\frac{\epsilon e n}{2^9 B}\right)} e^{-\epsilon^2 n/512B^2} = (**)$$

For big enough n $(**)$ is less than 1, which means that there is a sample $\mathbf{x} \in \mathcal{X}^n$ such that for all kernels $K, \tilde{K}$ we have $|D_1^{\mathbf{x}}(K, \tilde{K}) - D_D(K, \tilde{K})| \leq \epsilon/2$. More precisely, n should be bigger than $c d_\phi^2 B^5/\epsilon^5$ for some constant c. □

Now we can prove Lemma 3:

Proof of lemma 3. Fix some set of probability distributions $\mathbf{D} = (D_1, \ldots, D_T)$. For every D_t denote a sample described by Lemma 4 by $\mathbf{x}_t$. Let $\tilde{\mathcal{K}}$ be an $\epsilon/2T$-cover of $\mathcal{K}$ with respect to $D_1^{\mathbf{x}}$, where $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T) \in \mathcal{X}^{Tn}$ and $n = c d_\phi^2 B^5/\epsilon^5$. Then the following chain of inequalities holds:

$$\max_{h \in \mathcal{H}_K} \min_{h' \in \mathcal{H}_{\tilde{K}}} \frac{1}{Tn} \sum_{t=1}^{T} \sum_{i=1}^{n} |h(x_i^t) - h'(x_i^t)| \leq \max_{h} \min_{h'} \|h(\mathbf{x}) - h'(\mathbf{x})\| \leq$$

$$\max_{w} ||K_{\mathbf{x}}^{\frac{1}{2}} w - \tilde{K}_{\mathbf{x}}^{\frac{1}{2}} w|| \leq \|K_{\mathbf{x}}^{\frac{1}{2}} - \tilde{K}_{\mathbf{x}}^{\frac{1}{2}}\|_2 \leq \sqrt{\|K_{\mathbf{x}} - \tilde{K}_{\mathbf{x}}\|_2} \leq \sqrt{Tn |K_{\mathbf{x}} - \tilde{K}_{\mathbf{x}}|_\infty}.$$

Consequently, by Lemma 3 in [89]:

$$|\tilde{\mathcal{K}}| \leq N(\epsilon/2T, \mathcal{K}, D_1^{\mathbf{x}}) \leq \left(\frac{4en^3 T^5 B^2}{\epsilon^2 d_\phi} \right)^{d_\phi} = \left(C T^5 d_\phi^5 \left(\frac{B}{\epsilon} \right)^{17} \right)^{d_\phi} \tag{B.10}$$

. It is left to show that $\tilde{\mathcal{K}}$ is an ϵ-cover of $\mathcal{K}$ with respect to $D_{\mathbf{D}}$. By definition, for every $K \in \mathcal{K}$ there exists $\tilde{K} \in \tilde{\mathcal{K}}$ such that $D_1^{\mathbf{x}}(K, \tilde{K}) < \epsilon/2T$. Therefore for every $t = 1 \dots T$:

$$\max_{h \in \mathcal{H}_K} \min_{h' \in \mathcal{H}_{\tilde{K}}} \frac{1}{n} \sum_{i=1}^{n} |h(x_i^t) - h'(x_i^t)| \leq \max_{h \in \mathcal{H}_K} \min_{h' \in \mathcal{H}_{\tilde{K}}} \frac{T}{Tn} \sum_{t,i} |h(x_i^t) - h'(x_i^t)| < \frac{\epsilon}{2}.$$

Consequently, by Lemma 4, $D_{D_t}(K, \tilde{K}) < \epsilon$ for all $t = 1, \dots, T$. □

The statement of Theorem 10 follows from a combination of equation (B.4), its equivalent in the opposite direction with Theorem 19 and Lemmas 1 and 3.

B.2 Proof of Theorem 11

To prove Theorem 11 we introduce an intermediate quantity that can be seen as an *expected multi-task risk*:

$$\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) = \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{h\sim Q_t(x,y)\sim D_t} \ell(h(x),y). \tag{B.11}$$

First we will bound the uncertainty on the task environment level by bounding the difference between the expected lifelong error, $\mathrm{er}_{\mathfrak{D}}(\mathcal{Q})$, and expected multi-task error on the observed tasks, $\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q})$. Then we will bound the uncertainty within observed tasks by bounding the difference between $\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q})$, and its empirical approximation, $\widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q})$. Our main tool in both cases will be the following lemma.

Lemma 5. *Let f be a random variable taking values in A and let $X_1,\ldots,X_l$ be l independent random variables with each X_k distributed according to μ_k over the set A_k. For functions $g_k : A\times A_k \to [a_k,b_k],\ k=1\ldots l$, let $\xi_k(f) = \mathbb{E}_{X_k\sim\mu_k} g_k(f,X_k)$ for any fixed value of f. Then for any fixed distribution π on A and any $\lambda,\delta>0$ the following inequality holds with probability at least $1-\delta$ (over sampling $X_1,\ldots,X_l$) for all distributions ρ over A*

$$\mathop{\mathbb{E}}_{f\sim\rho}\sum_{k=1}^{l}\xi_k(f) - \mathop{\mathbb{E}}_{f\sim\rho}\sum_{k=1}^{l} g_k(h,X_k) \le \frac{1}{\lambda}\left(\mathrm{KL}(\rho||\pi) + \frac{\lambda^2}{8}\sum_{k=1}^{l}(b_k-a_k)^2 - \log\delta\right).$$

Proof. We start in a standard way by applying the change of measure inequality to $g(f) = \sum_{k=1}^{l}\xi_k(f) - \sum_{k=1}^{l} g_k(f,X_k)$"

$$\mathop{\mathbb{E}}_{f\sim\rho}\left(\sum_{k=1}^{l}\xi_k(f) - \sum_{k=1}^{l} g_k(f,X_k)\right) \le \frac{1}{\lambda}\left(\mathrm{KL}(\rho\|\pi) + \log \mathop{\mathbb{E}}_{f\sim\pi} e^{\lambda g(f)}\right). \tag{B.12}$$

Note, that

$$e^{\lambda g(f)} = \prod_{k=1}^{l}\exp(\lambda(\xi_k(f) - g_k(f,X_k))), \tag{B.13}$$

since for any fixed f the factors are independent. This allows us to apply Hoeffding's

lemma to each factor:

$$\mathop{\mathbb{E}}_{X_1\sim\mu_1}\cdots\mathop{\mathbb{E}}_{X_l\sim\mu_l} e^{\lambda g(f)} \le \exp\Big(\frac{\lambda^2}{8}\sum_{k=1}^{l}(b_k-a_k)^2\Big). \tag{B.14}$$

By taking the expectation over $f\sim\pi$ we obtain

$$\mathop{\mathbb{E}}_{f\sim\pi}\mathop{\mathbb{E}}_{X_1\sim\mu_1}\cdots\mathop{\mathbb{E}}_{X_l\sim\mu_l} e^{\lambda g(f)} \le \exp\Big(\frac{\lambda^2}{8}\sum_{k=1}^{l}(b_k-a_k)^2\Big). \tag{B.15}$$

Since π is fixed and does not depend on $X_1,\ldots,X_l$, we can exchange the order of expectations. By applying Markov's inequality with respect to expectations over $X_1,\ldots,X_l$ we obtain that with probability at least $1-\delta$:

$$\log \mathop{\mathbb{E}}_{f\sim\pi} e^{\lambda g(f)} \le \frac{\lambda^2}{8}\sum\nolimits_{k=1}^{l}(b_k-a_k)^2 - \log\delta. \tag{B.16}$$

We obtain (5) by combining (B.16) and (B.12). □

In order to bound the difference between $\mathrm{er}_{\mathfrak{D}}(\mathcal{Q})$ and $\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q})$ we treat each task t with the corresponding training sample S_t as a random variable and apply Lemma 5. Formally, we set $\rho=\mathcal{Q}$, $\pi=\mathcal{P}$, $X_k=(t_k,S_k)$, $l=T$, $f=P$ and $g_k(f,X_k)=\frac{1}{T}\mathop{\mathbb{E}}_{h\sim Q_k(x,y)\sim D_k} l(h(x),y)$ and apply Lemma 5 with $\lambda=\sqrt{T}$. Since $a_k=0$ and $b_k=\frac{1}{T}$ we obtain with probability at least $1-\delta/2$ that for all $\mathcal{Q}$

$$\mathrm{er}_{\mathfrak{D}}(\mathcal{Q}) \le \widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) + \frac{1}{\sqrt{T}}\left(\mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \frac{1}{8} - \log\frac{\delta}{2}\right). \tag{B.17}$$

To bound the difference between $\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q})$ and $\widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q})$ we apply Lemma 5 to the union of all training samples $S'=\bigcup_{t=1}^{T}S_t$. We set $\rho=(\mathcal{Q},Q_1,\ldots,Q_T)$, $\pi=(\mathcal{P},P,\ldots,P)$, $X_k=(x_i^t,y_i^t)$, $l=Tn$, $f=(P,h_1,\ldots,h_T)$ and $g_k(f,X_k)=\frac{1}{Tn}\ell(h_t(x_i^t),y_i^t)$. In this setting $a_k=0$ and $b_k=1/(Tn)$, Lemma 5 with $\lambda=T\sqrt{n}$ yields that with probability at least $1-\delta/2$ for all $\mathcal{Q}$

$$\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) \le \widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q}) + \frac{1}{T\sqrt{n}}\mathrm{KL}((\mathcal{Q},Q_1,\ldots,Q_T)||(\mathcal{P},P,\ldots,P)) + \frac{1}{8\sqrt{n}} - \frac{\log\delta/2}{T\sqrt{n}}. \tag{B.18}$$

The statement of Theorem 11 follows by a union bound from (B.17) and (B.18).

B.3 Proof of Theorem 12

We will prove Theorem 12 in the same way as we proved Theorem 11. In particular, note that conditioned on the observed tasks the corresponding training samples are independent, therefore we can reuse a step from the proof of Theorem 11 that bounds the difference between the expected multi-task risk $\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q})$ and the empirical error $\widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q})$ with probability at least $1-\delta$:

$$\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q}) + \frac{1}{T\sqrt{n}}\left(\mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \sum_{t=1}^{T} \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \mathrm{KL}(Q_t(P,S_t)||P)\right) + \frac{T + 8\log 1/\delta}{8T\sqrt{n}}. \tag{B.19}$$

To bound the difference between $\mathrm{er}_{\mathfrak{D}}(\mathcal{Q})$ and $\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q})$, however, we need a different argument that would take into account dependencies between the observed tasks:

Theorem 20. *For any fixed hyper-prior distribution $\mathcal{P}$, any proper exact fractional cover $\mathbf{C}$ of the dependency graph Γ of the observed T tasks of size k and any $\delta > 0$ the following holds with probability at least $1-\delta$ uniformly for all hyper-posterior distributions $\mathcal{Q}$:*

$$\mathrm{er}_{\mathfrak{D}}(\mathcal{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) + \sqrt{\frac{\mathbf{w}(\mathbf{C})}{n}}\,\mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \frac{\sqrt{\mathbf{w}(\mathbf{C})}(1 - 8\log\delta + 8\log k)}{8\sqrt{T}}. \tag{B.20}$$

Proof. By Donsker-Varadhan's variational formula:

$$\mathrm{er}_{\mathfrak{D}}(\mathcal{Q}) - \widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) = \sum_{j=1}^{k} \frac{w_j}{\mathbf{w}(\mathbf{C})} \mathop{\mathbb{E}}_{P\sim\mathcal{Q}} \frac{\mathbf{w}(\mathbf{C})}{T} \sum_{i\in C_j} \mathop{\mathbb{E}}_{(t,S_t)} \mathrm{er}_{D_t}(Q_t) - \mathrm{er}_{D_i}(Q_i) \leq \tag{B.21}$$

$$\sum_{j=1}^{k} \frac{w_j}{\mathbf{w}(\mathbf{C})\lambda_j}\left(\mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \log \mathop{\mathbb{E}}_{P\sim\mathcal{P}}\left(\exp\frac{\lambda_j \mathbf{w}(\mathbf{C})}{T} \sum_{i\in C_j} \mathop{\mathbb{E}}_{(t,S_t)} \mathrm{er}_{D_t}(Q_t) - \mathrm{er}_{D_i}(Q_i)\right)\right). \tag{B.22}$$

Since the tasks within every C_j are independent, by Hoeffding's lemma [38] for every fixed prior P, we have:

$$\mathop{\mathbb{E}}_{(t_i,S_i),i\in C_j} \exp\left(\frac{\lambda_j \mathbf{w}(\mathbf{C})}{T} \sum_{i\in C_j} \mathop{\mathbb{E}}_{(t,S_t)} \mathrm{er}_{D_t}(Q_t) - \mathrm{er}_{D_i}(Q_i)\right) \leq \exp\left(\frac{\lambda_j^2 \mathbf{w}(\mathbf{C})^2 |C_j|}{8T^2}\right). \tag{B.23}$$

Therefore, by Markov's inequality with probability at least $1-\delta_j$ it holds that:

$$\log \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \exp\left(\frac{\lambda_j \mathbf{w}(\mathbf{C})}{T}\sum_{i\in C_j}\mathop{\mathbb{E}}_{(t,S_t)} \mathrm{er}_{D_t}(Q_t)-\mathrm{er}_{D_i}(Q_i)\right) \leq \frac{\lambda_j^2 \mathbf{w}(\mathbf{C})^2|C_j|}{8T^2} - \log\delta_j.$$

Consequently, we obtain with probability at least $1-\sum_{j=1}^k \delta_j$:

$$\sum_{j=1}^k \frac{w_j}{\mathbf{w}(\mathbf{C})\lambda_j}\left(\mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \log \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \exp\frac{\lambda_j \mathbf{w}(\mathbf{C})}{T}\left(\sum_{i\in C_j}\mathop{\mathbb{E}}_{(t,S_t)} \mathrm{er}_{D_t}(Q_t)-\mathrm{er}_{D_i}(Q_i)\right)\right) \leq \tag{B.24}$$

$$\sum_{j=1}^k \frac{w_j}{\mathbf{w}(\mathbf{C})}\frac{1}{\lambda_j}\mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \sum_{j=1}^k \frac{w_j\lambda_j \mathbf{w}(\mathbf{C})|C_j|}{8T^2} - \sum_{j=1}^k \frac{w_j}{\mathbf{w}(\mathbf{C})\lambda_j}\log\delta_j. \tag{B.25}$$

By setting $\lambda_1 = \cdots = \lambda_k = \sqrt{\frac{T}{\mathbf{w}(\mathbf{C})}}$ and $\delta_j = \frac{w_j}{\mathbf{w}(\mathbf{C})}\delta$ we obtain the statement of the theorem. □

By combining (B.19) with Theorem 20 we obtain the statement of Theorem 12.

B.4 Proof of Theorem 13

Similarly to the previous section, we first bound the difference between $\widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q})$ and multi-task expected error given by:

$$\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) = \mathop{\mathbb{E}}_{A\sim\mathcal{Q}} \frac{1}{T-1}\sum_{t=2}^T \mathrm{er}_{D_t}(A). \tag{B.26}$$

The following theorem is a slight modification of the argument used to prove Theorems 11 and 12:

Theorem 21. *For any fixed hyper-prior distribution $\mathcal{P}$ with probability at least $1-\delta$ the following holds uniformly for all hyper-posterior distributions $\mathcal{Q}$:*

$$\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) \leq \widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q}) + \frac{1}{(T-1)\sqrt{n}}\mathrm{KL}(\mathcal{Q}\times Q_2\times\cdots\times Q_T||\mathcal{P}\times P_2\times\cdots\times P_T) + \frac{(T-1)+8\log 1/\delta}{8(T-1)\sqrt{n}}, \tag{B.27}$$

where $P_2, \ldots, P_T$ are some reference prior distributions that do not depend on the training sets of subsequent tasks. Possible choices include using just one prior distribution P fixed before observing any data, or using the posterior distributions obtained from the previous task, i.e. $P_t = Q_{t-1}$.

Proof. By applying KL-inequality we obtain:

$$\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q}) - \widehat{\mathrm{er}}_{\mathbf{S}}(\mathcal{Q}) \leq \frac{1}{\lambda}\Big(\mathrm{KL}(\mathcal{Q} \times Q_2 \times \cdots \times Q_T || \mathcal{P} \times P_2 \times \cdots \times P_T) + \log \mathop{\mathbb{E}}_{A\sim\mathcal{P}} \mathop{\mathbb{E}}_{h_2\sim P_2} \ldots \mathop{\mathbb{E}}_{h_T\sim P_T} \exp\Big(\frac{\lambda}{T-1}\sum_{t=2}^{T}\Big(\mathop{\mathbb{E}}_{(x,y)\sim D_t}\ell(h_t(x),y) - \frac{1}{n}\sum_{i=1}^{n}\ell(h(x_i^t),y_i^t)\Big)\Big)\Big).$$

Due to independence of any prior P_t from the consequent sample sets $S_t, \ldots, S_T$, we obtain that:

$$\mathop{\mathbb{E}}_{S_1,\ldots,S_T} \mathop{\mathbb{E}}_{A\sim\mathcal{P}} \mathop{\mathbb{E}}_{h_2\sim P_2} \ldots \mathop{\mathbb{E}}_{h_T\sim P_T} f_2(h_2,S_1)\cdot\cdots\cdot f_T(h_T,S_T) = \mathop{\mathbb{E}}_{A\sim\mathcal{P}} \mathop{\mathbb{E}}_{S_1} \mathop{\mathbb{E}}_{h_2\sim P_2} \mathop{\mathbb{E}}_{S_2} f_2(h_2,S_2) \ldots \mathop{\mathbb{E}}_{h_T\sim P_T} \mathop{\mathbb{E}}_{S_T} f_T(h_T,S_T),$$

where

$$f_t(h_t,S_t) = \frac{\lambda}{T-1}\Big(\mathop{\mathbb{E}}_{(x,y)\sim D_t}\ell(h_t(x),y) - \frac{1}{m}\sum_{i=1}^{n}\ell(h_t(x_i^t),y_i^t)\Big). \tag{B.28}$$

Due to Hoeffding's lemma, boundness of the loss and the fact that training samples are i.i.d., the following holds:

$$\mathop{\mathbb{E}}_{S_T} f_T(h_T,S_T) \leq \exp\Big(\frac{\lambda^2}{8(T-1)^2 n}\Big). \tag{B.29}$$

Therefore:

$$\mathop{\mathbb{E}}_{S_1,\ldots,S_T} \mathop{\mathbb{E}}_{A\sim\mathcal{P}} \mathop{\mathbb{E}}_{h_2\sim P_2} \ldots \mathop{\mathbb{E}}_{h_T\sim P_T} f_2(h_2,S_2)\cdot\cdots\cdot f_T(h_T,S_T) \leq \exp\Big(\frac{\lambda^2}{8(T-1)n}\Big). \tag{B.30}$$

By using Markov's inequality and setting $\lambda = (T-1)\sqrt{n}$ we obtain the statement of the theorem. Note, also, that the KL-term in this theorem can be simplified:

$$\mathrm{KL}(\mathcal{Q} \times Q_2 \times \cdots \times Q_T || \mathcal{P} \times P_2 \times \cdots \times P_T) = \mathrm{KL}(\mathcal{Q}||\mathcal{P}) + \sum_{t=2}^{T} \mathop{\mathbb{E}}_{A\sim\mathcal{Q}} \mathrm{KL}(Q_t||P_t).$$

□

To complete the proof of Theorem 13 we need to bound the difference between $\mathrm{er}(\mathcal{Q})$ and $\widehat{\mathrm{er}}_{\mathbf{D}}(\mathcal{Q})$. We start with proving the following result which is similar to Lemma 2 in [85]:

Lemma 6. *Let $X_1, \ldots, X_n \in \Omega$ be a sequence of random variables and $g : \Omega \to [0,1]$ be a function such that $\mathbb{E}[g(X_i)|X_1, \ldots, X_{i-1}] = b_i$. Let $Z_1, \ldots, Z_n$ be independent Bernoulli random variables such that $\mathbb{E}[Z_i] = b_i$. Then for any convex function f:*

$$\mathbb{E}[f(g(X_1), \ldots, g(X_n))] \leq \mathbb{E}[f(Z_1, \ldots, Z_n)]. \tag{B.31}$$

Proof. Any point $x = (x_1, \ldots, x_n) \in [0,1]^n$ can be written as a linear combination of the extreme points $\nu = (\nu_1, \ldots, \nu_n) \in \{0,1\}^n$ in the following way:

$$x = \sum_{\nu \in \{0,1\}^n} \left(\prod_{i=1}^{n} ((1 - x_i)(1 - \nu_i) + x_i \nu_i) \right) \nu. \tag{B.32}$$

Therefore by convexity of f we have that:

$$f(x) \leq \sum_{\nu \in \{0,1\}^n} \left(\prod_{i=1}^{n} ((1 - x_i)(1 - \nu_i) + x_i \nu_i) \right) f(\nu). \tag{B.33}$$

By taking expectations on both sides we obtain that:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{X_1^n} f(g(X_1), \ldots, g(X_n)) \leq \\
&\mathop{\mathbb{E}}_{X_1^n} \left[\sum_{\nu \in \{0,1\}^n} \left(\prod_{i=1}^{n} ((1 - g(X_i))(1 - \nu_i) + g(X_i)\nu_i) \right) f(\nu) \right] = \\
&\sum_{\nu \in \{0,1\}^n} \mathop{\mathbb{E}}_{X_1^n} \left[\prod_{i=1}^{n} ((1 - g(X_i))(1 - \nu_i) + g(X_i)\nu_i) \right] f(\nu) = \\
&\sum_{\nu \in \{0,1\}^n} \mathop{\mathbb{E}}_{X_1^{n-1}} \left[\mathop{\mathbb{E}}_{X_n} \left[\prod_{i=1}^{n} ((1 - g(X_i))(1 - \nu_i) + g(X_i)\nu_i) | X_1^{n-1} \right] \right] f(\nu) = \\
&\sum_{\nu \in \{0,1\}^n} \mathop{\mathbb{E}}_{X_1^{n-1}} \left[\left(\prod_{i=1}^{n-1} ((1 - g(X_i))(1 - \nu_i) + g(X_i)\nu_i) \right) \times \right. \\
&\left. \mathop{\mathbb{E}}_{X_n} [(1 - g(X_n))(1 - \nu_i) + g(X_n)\nu_i | X_1^{n-1}] \right] f(\nu) =
\end{aligned}$$

$$\sum_{\nu\in\{0,1\}^n} \mathop{\mathbb{E}}_{X_1^{n-1}} \left[\left(\prod_{i=1}^{n-1} ((1-g(X_i))(1-\nu_i) + g(X_i)\nu_i) \right) \times \right.$$
$$\left. ((1-b_n)(1-\nu_i) + b_n\nu_i) \right] f(\nu) = \ldots$$
$$\sum_{\nu\in\{0,1\}^n} \left(\prod_{i=1}^{n} ((1-b_i)(1-\nu_i) + b_i\nu_i) \right) f(\nu) = \mathop{\mathbb{E}}_{Z_1^n}[f(Z_1^n)].$$

□

We proceed by using techniques from [85] in combination of those from [77], resulting in the following lemma:

Lemma 7. *For any fixed algorithm A and any λ the following holds:*

$$\mathop{\mathbb{E}}_{E_1,\ldots,E_T} \exp\left(\lambda\left(\mathrm{er}(A) - \frac{1}{T-1}\sum_{t=2}^{T}\mathrm{er}_{D_t}(A)\right)\right) \leq \exp\left(\frac{\lambda^2}{2(T-1)}\right). \tag{B.34}$$

Proof. First, define $X_t = (E_{t-1}, E_t)$ for $t = 2,\ldots,T$ and $g : X_t \mapsto \mathrm{er}_{D_t}(A)$ and $b = \mathrm{er}(A)$. Then:

$$\exp\left(\lambda\left(\mathrm{er}(A) - \frac{1}{T-1}\sum_{t=2}^{T}\mathrm{er}_{D_t}(A)\right)\right)$$
$$= \exp\left(\frac{\lambda}{T-1}\left(\sum_{\text{even } t}(b-g(X_t)) + \sum_{\text{odd } t}(b-g(X_t))\right)\right)$$
$$\leq \frac{1}{2}\exp\left(\frac{2\lambda}{T-1}\sum_{\text{even } t}(b-g(X_t))\right) + \frac{1}{2}\exp\left(\frac{2\lambda}{T-1}\sum_{\text{odd } t}(b-g(X_t))\right). \tag{B.35}$$

Note, that both, the set of X_t-s corresponding to even t and the set of X_t-s corresponding to odd t, form a martingale difference sequence. Therefore by using Lemma 6 (or similarly Lemma 2 in [85]) and Hoeffding's lemma [38] we obtain:

$$\mathop{\mathbb{E}}_{E_1,\ldots,E_T} \exp\left(\frac{2\lambda}{T-1}\sum_{\text{even } t}(b-g(X_t))\right) \leq \exp\left(\frac{4\lambda^2}{8(T-1)}\right), \tag{B.36}$$

$$\mathop{\mathbb{E}}_{E_1,\ldots,E_T} \exp\left(\frac{2\lambda}{T-1}\sum_{\text{odd } t}(b-g(X_t))\right) \leq \exp\left(\frac{4\lambda^2}{8(T-1)}\right). \tag{B.37}$$

Together with inequality (B.35) this gives the statement of the lemma. □

Now we can prove the following statement:

Theorem 22. *For any hyper-prior distribution $\mathcal{P}$ and any $\delta > 0$ with probability at least $1 - \delta$ the following inequality holds uniformly for all $\mathcal{Q}$:*

$$\operatorname{er}(\mathcal{Q}) \leq \widehat{\operatorname{er}}_{\mathbf{D}}(\mathcal{Q}) + \frac{1}{\sqrt{T-1}} \operatorname{KL}(\mathcal{Q}||\mathcal{P}) + \frac{1 + 2\log 1/\delta}{2\sqrt{T-1}}. \tag{B.38}$$

Proof. By applying Donsker-Varadhan's variational formula [28] one obtains that:

$$\operatorname{er}(\mathcal{Q}) - \widehat{\operatorname{er}}_{\mathbf{D}}(\mathcal{Q}) \leq \frac{1}{\lambda} \left(\operatorname{KL}(\mathcal{Q}||\mathcal{P}) + \log \mathop{\mathbb{E}}_{A \sim \mathcal{P}} \exp \lambda \left(\operatorname{er}(A) - \frac{1}{T-1} \sum_{t=2}^{T} \operatorname{er}_{D_t}(A) \right) \right).$$

For a fixed algorithm A we obtain from Lemma 7:

$$\mathop{\mathbb{E}}_{E_1,\ldots,E_T} \exp \left(\lambda \left(\operatorname{er}(A) - \frac{1}{T-1} \sum_{t=2}^{T} \operatorname{er}_{D_t}(A) \right) \right) \leq \exp \left(\frac{\lambda^2}{2(T-1)} \right). \tag{B.39}$$

Since $\mathcal{P}$ does not depend on the process, by Markov's inequality, with probability at least $1 - \delta$, we obtain

$$\mathop{\mathbb{E}}_{A \sim \mathcal{P}} \exp \lambda \left(\operatorname{er}(A) - \frac{1}{T-1} \sum_{t=2}^{T} \operatorname{er}_{D_t}(A) \right) \leq \frac{1}{\delta} \exp \frac{\lambda^2}{2(T-1)}. \tag{B.40}$$

The statement of the theorem follows by setting $\lambda = \sqrt{T-1}$. □

By combining Theorems 21 and 22 we obtain Theorem 13.

B.5 Proofs of Theorems 14 and 15

We start with presenting the following two lemmas that show how to control the error propagation of the learned representations (sets of base classifiers).

Lemma 8. *Let* $V = MV(h_1, \ldots, h_k, g)$ *and* $\tilde{V} = MV(h_1, \ldots, h_k, \tilde{g})$*. Then, for any distribution* D*:*

$$d_D(V, \tilde{V}) \leq d_D(g, \tilde{g}). \tag{B.41}$$

Proof. By the definition of $d_D(V, \tilde{V})$ there exists $u \in V$ such that:

$$d_D(V, \tilde{V}) = d_D(u, \tilde{V}). \tag{B.42}$$

We can represent u as $u = \text{sign}(\sum_{i=1}^{k} \alpha_i h_i + \alpha g)$ and let $u_1 = \sum_{i=1}^{k} \alpha_i h_i$. Note that while all h_i-s, g and $\tilde{g}$ are assumed to take values in $\{-1, 1\}$, u_1 can take values in $\mathbb{R}$. Then:

$$\begin{aligned} d_D(u, \tilde{V}) &= \min_{\tilde{h} \in \tilde{V}} d_D(u, \tilde{h}) \leq \min_{\tilde{h} \in MV(u_1, \tilde{g})} d_D(u, \tilde{h}) \\ &\leq \max_{h \in MV(u_1, g)} \min_{\tilde{h} \in MV(u_1, \tilde{g})} d_D(h, \tilde{h}) = d_D(MV(u_1, g), MV(u_1, \tilde{g})). \end{aligned}$$

Now we show that for any $\alpha_1 u_1 + \alpha_2 g \in MV(u_1, g)$ there exists a close hypothesis in $MV(u_1, \tilde{g})$. In particular, this hypothesis is $\alpha_1 u_1 + \alpha_2 \tilde{g}$:

$$\begin{aligned} d_D(\alpha_1 u_1 + \alpha_2 g, \alpha_1 u_1 + \alpha_2 \tilde{g}) = \\ \mathbb{E}_{x \sim D} [\![\text{sign}(\alpha_1 u_1(x) + \alpha_2 g(x)) \neq \text{sign}(\alpha_1 u_1(x) + \alpha_2 \tilde{g}(x))]\!] = \\ \mathbb{E}_{x \sim D} [\![\alpha_1^2 u_1^2(x) + \alpha_1 \alpha_2 u_1(x) g(x) + \alpha_1 \alpha_2 u_1(x) \tilde{g}(x) + \alpha_2^2 g(x) \tilde{g}(x) < 0]\!]. \end{aligned}$$

Note that for every x on which g and $\tilde{g}$ agree, i.e. $g(x)\tilde{g}(x) = 1$, we obtain:

$$\begin{aligned} \alpha_1^2 u_1^2(x) + \alpha_1 \alpha_2 u_1(x) g(x) + \alpha_1 \alpha_2 u_1(x) \tilde{g}(x) + \alpha_2^2 g(x) \tilde{g}(x) \\ = (\alpha_1 u_1(x) + \alpha_2 g(x))^2 \geq 0. \end{aligned}$$

Therefore:

$$d_D(\alpha_1 u_1 + \alpha_2 g, \alpha_1 u_1 + \alpha_2 \tilde{g}) \leq \mathbb{E}_{x \sim D} [\![g(x) \neq \tilde{g}(x)]\!] = d_D(g, \tilde{g}). \tag{B.43}$$

□

Lemma 9. *Let* $V_k = MV(h_1, \ldots, h_k)$ *and* $\tilde{V}_k = MV(\tilde{h}_1, \ldots, \tilde{h}_k)$*. For any distribution* D*, if* $d_D(h_i, \tilde{h}_i) \leq \epsilon_i$ *for every* $i = 1, \ldots, k$*, then* $d_D(V_k, \tilde{V}_k) \leq \sum_{i=1}^{k} \epsilon_i$*.*

Proof. We will prove the statement by induction on k over a stronger statement that the conclusion holds for $V_k = MV(w_1, \ldots, w_l, h_1, \ldots, h_k)$ and $\tilde{V}_k = MV(w_1, \ldots, w_l, \tilde{h}_1, \ldots, \tilde{h}_k)$ for any $w_1, \ldots, w_l$. Note that for $k = 1$ the statement follows from Lemma 8.

Let $V'_k = MV(w_1, \ldots, w_l, h_1, \ldots, h_{k-1}, \tilde{h}_k)$. Then:

$$\begin{aligned} d_D(V_k, \tilde{V}_k) \leq & d_D(V_k, V'_k) + d_D(V'_k, \tilde{V}_k) \text{ (by triangular inequality)} \\ \leq & d_D(h_k, \tilde{h}_k) + d_D(V'_k, \tilde{V}_k) \text{ (by Lemma 8)} \\ \leq & \epsilon_k + \sum_{i=1}^{k-1} \epsilon_i \text{ (by assuption and induction).} \end{aligned}$$

□

B.5.1 Proof of Theorem 14

1. First, note that for every task Algorithm 3 solves at most 2 estimation problems with a probability of failure δ' for each of them. Therefore, with a union bound argument, the probability of any of these estimations being wrong is at most $2 \cdot T \cdot \delta' = \delta$. Thus, from now we assume that all the estimations were correct, that is, the high probability events of Theorem 2 hold.

2. To see that the error of every encountered task is bounded by ϵ, note that there are two cases. For a task t that is solved by a majority vote over previous tasks, we have $\widehat{\mathrm{er}}_{S_t}(g_t) + \sqrt{\widehat{\mathrm{er}}_{S_t}(g_t) \cdot \Delta_t} + \Delta_t \leq \epsilon$. In this case, Equation 2.8 in Theorem 2 implies $\mathrm{er}_{D_t, h_t^*}(g_t) \leq \epsilon$. For a task t that is not solved as a majority vote over previous tasks, we have $\Delta_t = \Delta(\mathrm{VC}(\mathcal{H}), \delta', |S_t|) \leq \epsilon/8k$. Since task t is realizable by the base class $\mathcal{H}$, we have $\inf_{h \in \mathcal{H}} \mathrm{er}_{D_t, h_t^*}(h) = 0$, and thus Equation 2.10 of Theorem 2 implies $\mathrm{er}_{D_t, h_t^*}(g_t) \leq \epsilon/8k < \epsilon$.

3. To upper bound the sample complexity we first prove that the number $\tilde{k}$ of tasks, which are not learned as majority votes over previous tasks, is at most k. For that we

use induction showing that for every $\hat{k} \leq \tilde{k}$, when we create a new $\tilde{h}_{\hat{k}}$ from the $i_{\hat{k}}$-th task, we have that

$$d_{D_{i_{\hat{k}}}}(h^*_{i_{\hat{k}}}, MV(h^*_{i_1}, \ldots, h^*_{i_{\hat{k}-1}})) > \gamma. \tag{B.44}$$

This implies $\tilde{k} \leq k$ by invoking that the γ-effective dimension of the sequence of encountered tasks is at most k.

To proceed to the induction, note that for $\hat{k} = 1$, the claim follows immediately. Consider $\hat{k} > 1$. If we create a new $\tilde{h}_{\hat{k}}$, it means that the condition in line 9 is true, which is:

$$\widehat{\mathrm{er}}_{S_{i_{\hat{k}}}}(g_{i_{\hat{k}}}) + \sqrt{\widehat{\mathrm{er}}_{S_{i_{\hat{k}}}}(g_{i_{\hat{k}}}) \cdot \Delta_i} + \Delta_i > \epsilon. \tag{B.45}$$

Therefore $\widehat{\mathrm{er}}_{S_{i_{\hat{k}}}}(g_{i_{\hat{k}}}) > 0.83\epsilon$. Consequently, due to (2.9), $\mathrm{er}_{D_{i_{\hat{k}}}, h^*_{i_{\hat{k}}}}(g_{i_{\hat{k}}}) > 0.67\epsilon$. Finally, by (2.10), $\inf_g \mathrm{er}_{D_{i_{\hat{k}}}, h^*_{i_{\hat{k}}}}(g) > 0.5\epsilon$. Therefore there is no majority vote predictor based on $\tilde{h}_1, \ldots, \tilde{h}_{\hat{k}-1}$ that leads to error less than $\epsilon/2$ on the problem $i_{\hat{k}}$. In other words:

$$d_{D_{i_{\hat{k}}}}(h^*_{i_{\hat{k}}}, MV(\tilde{h}_1, \ldots, \tilde{h}_{\hat{k}-1})) > \epsilon/2. \tag{B.46}$$

Now, by way of contradiction, suppose that $d_{D_{i_{\hat{k}}}}(h^*_{i_{\hat{k}}}, MV(h^*_{i_1}, \ldots, h^*_{i_{\hat{k}-1}})) \leq \gamma$. By construction for every $j = 1, \ldots, \hat{k} - 1$ $d_{D_{i_j}}(h^*_{i_j}, \tilde{h}_j) \leq \epsilon' \leq \epsilon/8k$. By the definition of discrepancy and the assumption on the marginal distributions it follows that for all j:

$$d_{D_{i_{\hat{k}}}}(h^*_{i_j}, \tilde{h}_j) \leq d_{D_{i_j}}(h^*_{i_j}, \tilde{h}_j) + \mathrm{disc}_H(D_{i_j}, D_{i_{\hat{k}}}) \leq \epsilon' + \xi. \tag{B.47}$$

Therefore by Lemma 9:

$$d_{D_{i_{\hat{k}}}}(MV(h^*_{i_1}, \ldots, h^*_{i_{\hat{k}-1}}), MV(\tilde{h}_1, \ldots, \tilde{h}_{\hat{k}})) \leq k(\epsilon' + \xi). \tag{B.48}$$

Consequently, by using the triangle inequality:

$$d_{D_{i_{\hat{k}}}}(h^*_{i_{\hat{k}}}, MV(\tilde{h}_1, \ldots, \tilde{h}_{\hat{k}-1})) \leq \gamma + k(\epsilon' + \xi) \leq \epsilon/4 + \epsilon/8 + \epsilon/8 = \epsilon/2, \tag{B.49}$$

which is in contradiction with (B.46).

4. The total sample complexity of Algorithm 3 consists of two parts. First, for every task Algorithm 3 checks, whether it can be solved by a majority vote over the base, at most $\tilde{k}$ predictors. VC-dimension of this class is $O(\tilde{k} \log \tilde{k} \log(\tilde{k} \log \tilde{k}))$ (Lemma 10.3

in [86]). For that it employs Theorem 2 and therefore needs the following number of samples:

$$\tilde{O}\left(\frac{T\tilde{k}\log\tilde{k}\log(\tilde{k}\log\tilde{k})\log(2T/\delta)}{\epsilon}\right)=\tilde{O}\left(\frac{Tk}{\epsilon}\right). \tag{B.50}$$

Second, there are at most $\tilde{k}$ tasks that do not satisfy the condition in line 9 and are learned using the hypothesis set $\mathcal{H}$ with estimation error $\epsilon'=\epsilon/(8k)$. Therefore the corresponding sample complexity is: $O\left(\frac{\tilde{k}\,\mathrm{VC}(\mathcal{H})\log(2T/\delta)}{\epsilon/(8k)}\right)=\tilde{O}\left(\frac{\mathrm{VC}(\mathcal{H})k^2}{\epsilon}\right)$.

B.5.2 Proof of Theorem 15

1. First, as in the proof of Theorem 14, we need to control the total probability of any conclusion of Algorithm 4 being incorrect. For every task $t=2,\ldots,T$ Algorithm 4 preforms at most two estimations. Therefore the total probability of failure is:

$$\delta_1+\sum_{t=2}^{T}2\delta_t=\frac{\delta}{2}+\sum_{l=1}^{\lfloor\log T\rfloor}2(2^{l+1}-2^l)\frac{\delta}{2^{2l+2}}=\frac{\delta}{2}+\frac{\delta}{2}\sum_{l=1}^{\lfloor\log T\rfloor}\frac{1}{2^l}\leq$$
$$\frac{\delta}{2}+\frac{\delta}{2}\sum_{l=1}^{\infty}\frac{1}{2^l}=\frac{\delta}{2}+\frac{\delta}{2}=\delta.$$

2. Performance guarantees follow from the design of the algorithm (as in Theorem 14).

3. The fact that $\tilde{k}\leq k$ can be proven in a way analogous to Theorem 14. However, we need to make sure that for every $\hat{k}=1,\ldots,\tilde{k}$, by using Lemma 9, we will obtain a suitable result. In particular, by construction for every $j=1,\ldots,\hat{k}-1$ $d_{D_{i_j}}(h^*_{i_j},\tilde{h}_j)\leq\epsilon'_j$. Therefore by Lemma 9

$$d_{D_{i_{\hat{k}}}}(MV(h^*_{i_1},\ldots,h^*_{i_{\hat{k}-1}}),MV(\tilde{h}_1,\ldots,\tilde{h}_{\hat{k}-1}))\leq(\hat{k}-1)\xi+\sum_{j=1}^{\hat{k}-1}\epsilon'_j. \tag{B.51}$$

By the definition of ϵ'_j:

$$\sum_{j=1}^{\hat{k}-1}\epsilon'_j\leq\frac{\epsilon}{16}+\sum_{m=1}^{\lfloor\hat{k}\rfloor}(2^{m+1}-2^m)\frac{\epsilon}{2^{2m+4}}=\frac{\epsilon}{16}+\frac{\epsilon}{16}\sum_{m=1}^{\lfloor\hat{k}\rfloor}\frac{1}{2^m}<\frac{\epsilon}{16}+\frac{\epsilon}{16}=\frac{\epsilon}{8}.$$

Together with the assumption on discrepancies, this guarantees that:

$$d_{D_{i_{\hat{k}}}}(MV(h^*_{i_1},\dots,h^*_{i_{\hat{k}-1}}), MV(\tilde{h}_1,\dots,\tilde{h}_{\hat{k}-1)})) \leq \frac{\epsilon}{4}, \tag{B.52}$$

which is exactly what is needed to come to contradiction.

4. The sample complexity of Algorithm 4 consists of the same parts as that of Algorithm 3.

The first difference comes from the fact that δ' changes over time, because the algorithm does not know the total number of tasks. However, the smallest value it attains is $\delta/(4T^2)$ and, since the dependence of the sample complexity on the δ is only logarithmic, it does not change the result significantly.

The second difference is that also ϵ' changes over time, because the algorithm does not know the parameter k in advance. This influences the sample complexity of learning "base tasks". In order to control it we need to control the following sum:

$$\sum_{j=1}^{\tilde{k}} \frac{1}{\epsilon'_j} \leq \sum_{m=1}^{\lfloor \log k \rfloor} (2^{m+1}-2^m)\frac{2^{2m+4}}{\epsilon} = \frac{16}{\epsilon}\sum_{m=1}^{\lfloor \log k \rfloor} 2^{3m} \leq \frac{k^3 \log k}{\epsilon}.$$

Therefore the complexity of learning the "base tasks" is:

$$\tilde{O}\left(\frac{\mathrm{VC}(\mathcal{H})k^3}{\epsilon}\right). \tag{B.53}$$

C Supplementary lemmas

Lemma 10 (Corollary 6.10 in [64]). *Let W_0^n be a martingale with respect to a sequence of random variables $(B_1, \ldots, B_n)$. Let $b_1^n = (b_1, \ldots, b_n)$ be a vector of possible values of the random variables $B_1, \ldots, B_n$. Let*

$$r_i(b_1^{i-1}) = \sup_{b_i}\{W_i : B_1^{i-1} = b_1^{i-1}, B_i = b_i\} - \inf_{b_i}\{W_i : B_1^{i-1} = b_1^{i-1}, B_i = b_i\}. \tag{C.1}$$

Let $r^2(b_1^n) = \sum_{i=1}^n (r_i(b_1^{i-1}))^2$ and $\widehat{R}^2 = \sup_{b_1^n} r^2(b_1^n)$. Then

$$\Pr_{B_1^n}\{W_n - W_0 > \epsilon\} < \exp\left(-\frac{2\epsilon^2}{\widehat{R}^2}\right). \tag{C.2}$$

Lemma 11 (Originally [38]; in this form Theorem 18 in [93]). *Let $\{U_1, \ldots, U_m\}$ and $\{W_1, \ldots, W_m\}$ be sampled uniformly from a finite set of d-dimensional vectors $\{v_1, \ldots, v_N\} \subset \mathbb{R}^d$ with and without replacement respectively. Then for any continuous and convex function $F : \mathbb{R}^d \to \mathbb{R}$ the following holds:*

$$\mathbb{E}\left[F\left(\sum_{i=1}^m W_i\right)\right] \leq \mathbb{E}\left[F\left(\sum_{i=1}^m U_i\right)\right] \tag{C.3}$$

Lemma 12 (Part of Lemma 19 in [93]). *Let $x = (x_1, \ldots, x_l) \in \mathbb{R}^l$. Then the following function is convex:*

$$F(x) = \sup_{i=1\ldots l} x_i. \tag{C.4}$$

Lemma 13 (Theorem 1 in [56]). *Let $X_1, \ldots, X_n$ be independent random variables taking values in the set $\mathcal{X}$ and f be a function $f : \mathcal{X}^n \to \mathbb{R}$. For any $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$ and $y \in \mathcal{X}$ define:*

$$\begin{aligned} x_{y,k} &= (x_1, \ldots, x_{k-1}, y, x_{k+1}, \ldots, x_n) \\ (\inf_k f)(x) &= \inf_{y \in \mathcal{X}} f(x_{y,k}) \\ \Delta_{+,f} &= \sum_{i=1}^{n} (f - \inf_k f)^2. \end{aligned}$$

Then for $t > 0$:

$$\Pr\{f - \mathbb{E}\, f \geq t\} \leq \exp\left(\frac{-t^2}{2\|\Delta_+\|_\infty}\right). \tag{C.5}$$