A Theoretical Framework for Deep Transfer Learning

Tomer Galanti
The School of Computer Science
Tel Aviv University
tomer22g@gmail.com

Lior Wolf
The School of Computer Science
Tel Aviv University
wolf@cs.tau.ac.il

Tamir Hazan
Faculty of Industrial Engineering & Management
Technion
tamir.hazan@gmail.com

Abstract

We generalize the notion of PAC learning to include transfer learning. In our framework, the linkage between the source and the target tasks is a result of having the sample distribution of all classes drawn from the same distribution of distributions, and by restricting all source and a target concepts to belong to the same hypothesis subclass. We have two models: an adversary model and a randomized model.

In the adversary model, we show that for binary classification, conventional PAC-learning is equivalent to the new notion of PAC-transfer and to transfer generalization of the VC-dimension. For regression, we show that PAC-transferability may exist even in the absence of PAC-learning. In the randomized model, we provide PAC-Bayesian and VC-style generalization bounds to transfer learning, including bounds specifically derived for Deep Learning. A wide discussion on the tradeoffs between the different involved parameters in the bounds is provided.

We demonstrate both cases in which transfer does not reduce the sample size ("trivial transfer") and cases in which the sample size is reduced ("non-trivial transfer").

1 Introduction

The advent of deep learning has helped promote the everyday use of transfer learning in a variety of learning problems. Representations, which are nothing more than activations of the network units at the deep layers, are used as general descriptors even though the network parameters were obtained while training a classifier on a specific set of classes under a specific sample distribution. As a result of the growing popularity of transferring deep learning representations, the need for a suitable theoretical framework has increased.

In the transfer learning setting that we consider, there are source tasks along with a target task. The source tasks are used to aid in the learning of the target task. However, the loss of the source tasks is not part of the learner's goal. As an illustrative example, consider the use of deep learning for the task of face recognition. There are 7 billion classes, each corresponding to a person, and each has its own indicator function (classifier). Moreover, the distribution of the images of each class is different. Some individuals are photographed more casually, while others are photographed in formal events. Some are photographed mainly under bright illumination, while the images of others are taken indoors. Hence, a complete discussion of transfer learning has to take into account both the classifiers and the distribution of the class samples.

A deep face-recognition neural-network is trained on a small subset of the classes. For example, the DeepFace network of Taigman et al. (2014) is trained using images of only 4030 persons. The activations of the network, at the layer just below the classification layer, are then used as a generic tool to represent any face, regardless of the image distribution of that person's album images.

In this paper, we study a transferability framework, which is constructed to closely match the theory of the learnable and its extensions including PAC learning (Valiant, 1984) and VC dimension. A fundamental Theorem of transfer learning, which links these concepts in the context of transfer learning, is provided. We introduce the notion of a simplifier that has the ability to return a subclass that is a good approximation of the original hypothesis class and is easier to learn. The conditions for the existence of a simplifier are discussed, and we show cases of transferability despite infinite VC dimensions. PAC-Bayesian and VC bounds are derived, in particular for the case of Deep Learning. A few illustrative examples demonstrate the mechanisms of transferability.

A cornerstone of our framework is the concept of a factory. Its role is to tie together the distributions of the source tasks and the target task without explicitly requiring the underlying distributions to be correlated or otherwise closely linked. The factory simply assumes that the distribution of the target task and the distributions of the source tasks are drawn i.i.d from the same distribution of distributions. In the face recognition example above, the subset of individuals used to train the network are a random subset of the population from which the target class (another individual) is also taken. The factory provides a subset of the population and a dataset corresponding to each person. The goal of the learner is to be able to learn efficiently how to recognize a new person's face using a relatively small dataset of the new person's face images. This idea generalizes the classic notion of learning in which the learner has access to a finite sample of examples and its goal is to be able to classify wisely a new unseen example.

Table 1: Summary of notations

ϵ, δ	error rate and confidence parameters $\in (0,1)$
$\mathcal{X}$	instances set
$\mathcal{Y}$	labels set
Z	examples set; usually $\mathcal{X} \times \mathcal{Y}$
p	a distribution
d	a task (a distribution over Z)
k	the number of source tasks
m	the number of samples for each source task
U	a finite set of distributions; usually $U = \{d_1, ..., d_k\}$ or $U = \{p_1, ..., p_k\}$
$\mathcal{E}'$	a set of distributions over $\mathcal{X}$
$\mathcal{E}$	an environment, a set of tasks
$prob_p(X)$ or $p(X)$	the probability of a set X in the distribution p
$\mathbb{P}, \mathbb{E}$	the probability and expectation operators
$\mathbb{P}[X\|Y], \mathbb{E}[X\|Y]$	the conditional probability and expectation
$\mathcal{D}[K]$ or just $\mathcal{D}$	a distribution over distributions (see Definitions 3, 4)
K	the subject of a factory
$s = \{z_1, ..., z_m\}$	data of m examples $\forall i : z_i \in Z$
$S = (s_{[1,k]}, s_t)$	k source data sets $s_1, ..., s_k$ (of same size) and one target data set s_t
$o = \{x_1, ..., x_m\}$	data of m instances $\forall i : x_i \in \mathcal{X}$
$O = (o_{[1,k]}, o_t)$	data of k of unlabeled source data sets $o_1, ..., o_k$ (of same size) and one target data set o_t
$S \sim \mathcal{D}[k, m, n]$	data set S according to the factory $\mathcal{D}$ with sizes $\forall i \in [k] : \|s_i\| = m$ and $\|s_t\| = n$
$S \sim \mathcal{D}[k, m]$	source data set S according to the factory $\mathcal{D}$ with sizes $\forall i \in [k] : \|s_i\| = m$
$U \sim \mathcal{D}[k]$	set of tasks of size k taken from $\mathcal{D}$
$d \sim \mathcal{D}$	a task taked from $\mathcal{D}$
$\mathcal{H}$	a hypothesis class (in the supervised case, a set of functions $\mathcal{X} \to \mathcal{Y}$)
c	a concept; an item of $\mathcal{H}$
$\mathscr{C}$	a hypothesis class family; a set of subsets in $\mathcal{H}$ such that $\mathcal{H} = \bigcup_{B \in \mathscr{C}} B$
B	a bias; i.e, $B \in \mathscr{C}$ (and $B \subset \mathcal{H}$)
N	an algorithm that outputs hypothesis classes
A	an algorithm that outputs concepts
$r(s)$	the application of an algorithm r on data s
$\ell : \mathcal{H} \times Z \to \mathbb{R}$	a loss function
0-1 loss	$\ell(c, (x, y)) = ([c(x) = y] = \text{true})$
squared loss	$\ell(c, (x, y)) = (c(x) - y)^2/2$
T	a learning setting; usually $T = (\mathcal{H}, Z, \ell)$
T_{PB}	a PAC-Bayes setting; usually $T_{TB} = (T, Q, p)$
$\mathscr{T}$	a transfer learning setting; usually $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$
$\epsilon_d(c)$	the generalization risk function = the expectation of $\ell(c, z)$, i.e, $\mathbb{E}_{z \sim d}[\ell(c, z)]$
$\epsilon_s(c)$	the empirical risk function; $\epsilon_s(c) = \frac{1}{\|s\|} \sum_{z \in s} \ell(c, z)$
$g : \mathscr{C} \times \mathcal{E} \to \mathbb{R}$	the infimum risk $g(B, d) = \inf_{c \in B} \epsilon_d(c) = \inf\{\epsilon_d(c) : c \in B\}$
$\epsilon_{\mathcal{D}}(B)$	transfer generalization risk = $\mathbb{E}_{d \sim \mathcal{D}}[g(B, d)]$
$\epsilon_U(B)$	source generalization risk = $\frac{1}{\|U\|} \sum_{d \in U}[g(B, d)]$
$\epsilon_s(B, r)$	2-step empirical risk = $\epsilon_s(r_B(s))$
$\epsilon_S(B, r)$	2-step source empirical risk = $\frac{1}{k} \sum_{i=1}^{k}[\epsilon_{s_i}(r_B(s_i))]$
$R(q)$	randomized transfer risk = $\mathbb{E}_{B \sim q}[\epsilon_{\mathcal{D}}(B)]$
$R_U(q)$	randomized source generalization risk = $\mathbb{E}_{B \sim q}[\epsilon_U(B)]$
$\mathrm{KL}(q\|p)$	KL-divergence, i.e, $\mathrm{KL}(q\|p) = \mathbb{E}_{x \sim q}[\log(q(x)/p(x))]$
$\epsilon_p(c_1, c_2)$	the mutual error rate; $\epsilon_p(c_1, c_2) = \epsilon_{(p, c_1)}(c_2)$
$\epsilon_o(c_1, c_2)$	the mutual empirical error rate; $\epsilon_o(c_1, c_2) = \epsilon_{c_1(o)}(c_2)$
$\mathrm{err}_p(B, K)$	the compatibility error rate; $\mathrm{err}_p(B, K) = \sup_{c_1 \in K} \inf_{c_2 \in B} \epsilon_p(c_1, c_2)$
$\mathrm{err}_o(B, K)$	the empirical compatibility error rate; $\mathrm{err}_o(B, K) = \sup_{c_1 \in K} \inf_{c_2 \in B} \epsilon_o(c_1, c_2)$

Table 2: Summary of notations (continued)

$E_U(B,K)$	the source compatibility error rate; $E_U(B,K) = \frac{1}{\lvert U\rvert}\sum_{p\in U}\mathrm{err}_p(B,K)$
$E(B,K)$	the generalization compatibility error rate; $E(B,K) = \mathbb{E}_{p\sim\mathcal{D}}[\mathrm{err}_p(B,K)]$
$E_O(B,K)$	the source empirical compatibility error rate; $E_O(B,K) = \frac{1}{\lvert O\rvert}\sum_{o\in O}\mathrm{err}_o(B,K)$
$h_{V,E,\sigma,w}$	a neural network with architecture (V,E,σ) and weights $w : E \to \mathbb{R}$
$\mathcal{H}_{V,E,\sigma}$	set of all neural networks with architecture (V,E,σ)
$\mathcal{H}^I_{V,E,\sigma}$	family of all subsets of $\mathcal{H}_{V,E,\sigma}$ determined by fixing weights on $I \subset E$
$E = I \cup J$	a set of edges in a neural network, I is the set of edges in the transfer architecture and J the rest of the edges (i.e, $I \cap J = \emptyset$)
$\mathcal{H}_{V,E,j,\sigma}$	the architecture induced by (V,E,σ) when taking only the first j layers (see Section 6)
$\mathrm{ERM}_B(s)$	empirical risk minimizer; $\mathrm{ERM}_B(s) = \arg\min_{c\in B}\epsilon_s(c)$
$\text{C-ERM}_{\mathscr{C}}(s_{[1,k]})$	class empirical risk minimizer; $\text{C-ERM}_{\mathscr{C}}(s_{[1,k]}) = \arg\min_{B\in\mathscr{C}}\frac{1}{k}\sum_{i=1}^{k}\min_{c\in B}\epsilon_{s_i}(c)$
$c^*_{i,B}$	empirical risk minimizer in B for the i'th data set; $c^*_{i,B} = \mathrm{ERM}_B(s_i)$
$r_{i,B}$	the application of a learner r_B of B on s_i; $r_{i,B} = r_B(s_i)$
$u\|\|v$	concatenation of the vectors u, v
0_s	a zeros vector of length s
$\mathbb{1}$	a unit matrix
$N_u(\epsilon,\delta)$	a universal bound on the sample complexity for learning any hypothesis class of VC dimension $\le u$
E_h	the set of all disks around 0 that lie on the hyperplane h
$\mathrm{vc}(\mathcal{H})$	the VC dimension of the hypothesis class $\mathcal{H}$
$\tau_{\mathcal{H}}(m)$	the growth function of the hypothesis class $\mathcal{H}$; i.e, $\tau_{\mathcal{H}}(m) = \max_{\{x_1,\ldots,x_m\}\in\mathcal{X}^m}\left\lvert\{(c(x_1),\ldots,c(x_m)) : c\in\mathcal{H}\}\right\rvert$
$\tau(k,m,r)$	the transfer growth function of the hypothesis class $\mathcal{H}$; i.e, $\tau(k,m,r) = \max_{\{s_1,\ldots,s_k\}\in\mathcal{Z}^{mk}}\left\lvert\{(r_{1,B}(s_1),\ldots,r_{k,B}(s_k)) : B\in\mathscr{C}\}\right\rvert$
$\tau(k,m;\mathscr{C},K)$	the adversary transfer growth function; i.e, $\tau(k,m;\mathscr{C},K) = \max_{\{o_1,\ldots,o_k\}\in\mathcal{X}^{mk}}$ $\left\lvert\{c_{1,1}(o_1),c_{1,2},\ldots,c_{k,1}(o_k),c_{k,2}(o_k)) : c_{i,1}\in K \text{ and } c_{i,2} = \mathrm{ERM}_B(c_{i,1}(o)) \text{ s.t } B\in\mathscr{C}\}\right\rvert$

2 Background

In this part, a brief introduction of the background required is provided. The general learning framework, the PAC-Bayesian setting and deep learning are introduced. These subjects are used and extended in this work. A reader who is familiar with these concepts, may skip to the next sections.

The general learning setting Recall the general learning setting proposed by Vapnik (1995). This setting generalizes classification, regression, multiclass classification, and several other learning settings.

Definition 1. *A learning setting $T = (\mathcal{H}, Z, \ell)$ is specified by,*

- *A hypothesis class $\mathcal{H}$.*
- *An examples set Z (with a sigma-algebra).*
- *And a loss function $\ell : \mathcal{H} \times Z \to \mathbb{R}$.*

This approach helps to define supervised learning settings such as binary classification and regression in a formal and very clean way. Furthermore, in this framework, one can define learning scenarios when the concepts are not functions of examples, but still have relations with examples from Z measured by loss functions (e.g, clustering, density estimation, etc.). If nothing else is mentioned, T stands for a learning setting. We say that T is learnable if the corresponding $\mathcal{H}$ Is learnable. In addition, if $\mathcal{H}$ has a VC dimension d, we say that T also has a VC dimension d. With these notions, we present an extended transfer learning setting, as a special case of the general learning setting with a few changes.

If a distribution d over Z is specified, the fitting of each $c \in \mathcal{H}$ is measured by a *Generalization Risk*,

$$\epsilon_d(c) = \mathbb{E}_{z\sim d}[\ell(c, z)]$$

Here, $\mathcal{H}$, Z and ℓ are known to the learner. The distribution d is called a *task* and is kept unknown. The goal of the learner is to pick $c \in \mathcal{H}$ that is closest to $\inf_{c\in\mathcal{H}} \epsilon_d(c)$. Since the distribution is unknown, this cannot be computed directly and only approximated using an empirical data set $\{z_1, ..., z_m\}$ selected i.i.d according to d. In many machine learning algorithms, the empirical risk function, $\epsilon_s(c) = \frac{1}{m}\sum_{z\in s} \ell(c, z)$ has great impact in the selection of the output hypothesis.

Binary classification: $Z = \mathcal{X} \times \{0, 1\}$ and $\mathcal{H}$ consisting of $c : \mathcal{X} \to \{0, 1\}$ with ℓ a 0-1 loss.

Regression: $Z = \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ are bounded subsets of $\mathbb{R}^n$ and $\mathbb{R}$ respectively. $\mathcal{H}$ is a set of bounded functions $c : \mathcal{X} \to \mathbb{R}$ and ℓ is any bounded function.

One of the early breakthroughs in statistical learning theory was the seminal work of Vapnik & Chervonenkis (1971) and the later work of Blumer et al. (1989), which characterized binary classification settings as learnable if and only if the VC dimension is finite. The VC dimension is the largest size required to ensure that there is a set of examples (of that size) such that any configuration of labels on it is consistent with one of the functions in $\mathcal{H}$.

Their analysis was based on the growth function,

$$\tau_{\mathcal{H}}(m) = \max_{o\in\mathcal{X}^m} \Big|\{c(x_1), ..., c(x_m) : c \in \mathcal{H}\}\Big|, \text{ where } o = \{x_1, ..., x_m\}$$

A famous Lemma due to Sauer (1972) asserts that whenever the VC dimension of the hypothesis class $\mathcal{H}$ is finite, then the growth function is polynomial in m,

$$\tau_{\mathcal{H}}(m) \leq \left(\frac{em}{\mathrm{vc}(\mathcal{H})}\right)^{\mathrm{vc}(\mathcal{H})} \quad \text{when } m > \mathrm{vc}(\mathcal{H}) \tag{1}$$

Theorem 1 (Vapnik & Chervonenkis (1971))**.** *Let d be any distribution over an examples set Z, $\mathcal{H}$ a hypothesis class and $\ell : \mathcal{H} \times Z \to \{0, 1\}$ be the 0-1 loss function. Then*

$$\mathbb{E}_{s\sim d^m}\left[\sup_{c\in\mathcal{H}} \Big|\epsilon_d(c) - \epsilon_s(c)\Big|\right] \leq \frac{4 + \sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\sqrt{2m}}$$

In particular, whenever the growth function is polynomial then the generalization risk and empirical risk uniformly converge to each other.

PAC-Bayes setting The PAC-Bayesian bound due to McAllester (1998) describes the *Expected Generalization Risk* (or simply expected risk), i.e, the expectation of the generalization risk with respect to a distribution over the hypothesis class. The aim is not measuring the fitting of each hypothesis directly but to measure the fitting of different distributions (perturbations) over the hypothesis class. The expected risk is measured by $\mathbb{E}_{c\sim q}[\epsilon_d(c)]$ and the *Expected Empirical Risk* is $\mathbb{E}_{c\sim q}[\epsilon_s(c)]$, where $s = \{z_1, ..., z_m\}$ (satisfying $\mathbb{E}_{s\sim d^m}\mathbb{E}_{c\sim q}[\epsilon_s(c)] = \mathbb{E}_{c\sim q}[\epsilon_d(c)]$). The PAC-Bayes bound estimates the expected risk with the expected empirical risk and a penalty term which decreases as the size of the training data set grows. A prior distribution p dictating a hierarchy between the hypotheses in $\mathcal{H}$ is selected. The PAC-Bayesian bound penalizes the posterior selection of q by the relative entropy between q and p, measured by the Kullback-Leibler divergence.

Definition 2 (PAC-Bayes setting)**.** *A PAC-Bayes setting $T_{PB} = (T, \mathcal{Q}, p)$ is specified by,*

- *A learning setting $T = (\mathcal{H}, Z, \ell)$.*
- *A set $\mathcal{Q}$ of posterior distributions q over $\mathcal{H}$.*
- *A prior distribution p over $\mathcal{H}$.*
- *The loss ℓ is bounded in* $[0, 1]$.

There are many variations of the PAC-Bayesian bound. Each of which has its own properties and advantages. In this work we refer to the original bound due to McAllester (1998).

Theorem 2 (McAllester (1998))**.** *Let d be any distribution over an example set Z, $\mathcal{H}$ a hypothesis class and $\ell : \mathcal{H} \times Z \to [0, 1]$ be a loss function. Let p be a distribution over $\mathcal{H}$ and $\mathcal{Q}$ a family of distributions over $\mathcal{H}$. Let $\delta \in (0, 1)$, then*

$$\mathbb{P}_{s\sim d^m}\left[\forall q \in \mathcal{Q} : \mathbb{E}_{c\sim q}[\epsilon_d(c)] \leq \mathbb{E}_{c\sim q}[\epsilon_s(c)] + \sqrt{\frac{\mathrm{KL}(q\|p) + \log(m/\delta)}{2(m-1)}}\right] \geq 1 - \delta$$

Where, $\mathrm{KL}(q\|p) = \mathbb{E}_{c\sim q}[\log(q(c)/p(c))]$.

Deep learning A neural network architecture (V, E, σ) is determined by a set of neurons V, a set of directed edges E and an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. In addition, a neural network of a certain architecture is specified by a weight function $w : E \to \mathbb{R}$. We denote $\mathcal{H}_{V,E,\sigma}$ the hypothesis class consisting of all neural networks with architecture (V, E, σ).

In this work we will only consider feedforward neural networks, i.e., those with no directed cycles. In such networks, the neurons are organized in disjoint layers, $V_0, ..., V_N$, such that $V = \bigcup_{i=1}^{N} V_i$. These functions have an output layer V_N consisting of only one neuron and input layer V_0 holding the input and one constant neuron that always hold the value 1. The other layers are called hidden. A fully connected neural network is a neural network in which every neuron of layer V_i is connected to every neuron of layer V_{i+1}. The computation done in feedforward neural networks is as follows: each neuron takes the outputs $(x_1, ..., x_h)$ of the neurons connected to it from the previous layer and the weights on the edges connecting between them $(w_1, ..., w_h)$ and outputs: $\sigma\left(\sum_{i=1}^{h} w_i \cdot x_i\right)$, see Figure 1. The output of the entire network is the value produced by the output neuron, see Figure 2.

In this paper we give special attention for the sign activation function that returns -1 if the input is negative and 1 elsewise. The reason is that such neural networks are very expressive and are easier to analyse. Such networks define compound functions of half-spaces.

Before we move on to sections dealing with general purpose transferability and the special case of deep learning, we would like to give some insights on our interpretation of common knowledge within neural networks. The classic approach to transfer learning in deep learning is done by shared weights. Concretely, some weights are shared between neural networks of similar architectures, each solving a different task. We adopt the following notation,

$$\mathcal{H}^{I}_{V,E,\sigma} = \{B_u |\ u : I \to \mathbb{R}\}\,, \quad \text{s.t} \ \ B_u = \{h_{V,E,\sigma,w} \mid \forall e \in I : w(e) = u(e)\} \ \text{ and } \ I \subset E$$

to denoe a family of subclasses of the hypothesis class $\mathcal{H}_{V,E,\sigma}$, each determined by a fixing of the weights on the edges in $I \subset E$. We will also denote by J the complement (i.e, $I \cup J = E$ and $I \cap J = \emptyset$). This will be a cornersote in formulating shared parameters between neural networks in

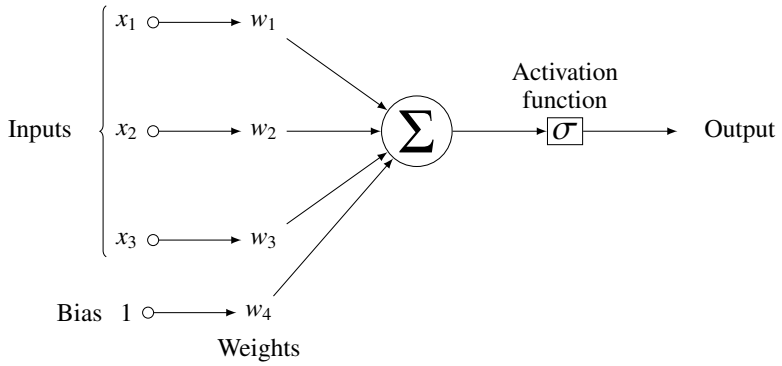

Figure 1: **A neuron:** four input values; $x_1, x_2, x_3, 1$, weights; w_1, w_2, w_3, w_4 and σ activation function.

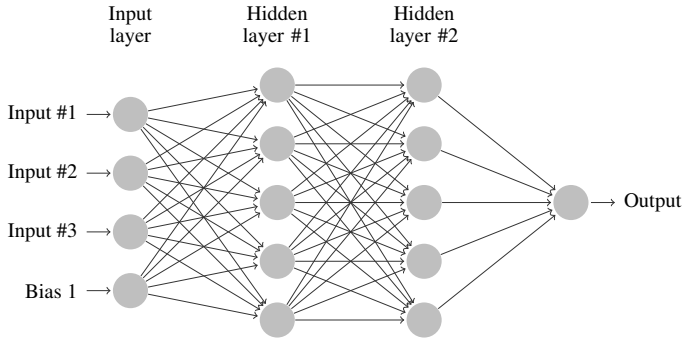

Figure 2: **A neural network:** feedforward fully connected neural network with four input neurons and two hidden layers, each containing five neurons.

transfer learning. For each B_u, every two neural networks $h_1, h_2 \in B_u$ share the same weights u on the edges in $I \subset E$, see Figure 3.

In most practical cases the activation of a neuron is determined by activations from the previous layers by a set of edges that are either in I are do not intersect I. However, in this paper, for the PAC-Bayes setting of deep learning, the discussion is kept more general, and activations can be determined by both transfered weights and non-transfered weights.

For VC-type bounds, the discussion is limited to the common situation in which the architecture is decomposed into two parts: the *transfer architecture* and the *specific architecture*, i.e,

$$h_{V,E,\sigma,u\|v} = h_2 \circ h_1$$

Where h_1 is a neural network consisting of the first j layers and the edges between them (with potentially more than one output) and h_2 has h_1's output as input and produces the one output of the whole network. With the previous notions, this tends to be the case where I consists of all edges between the first j layers, see Figure 4.

In this case, the family of hypothesis classes $\mathcal{H}^I_{V,E,\sigma}$ is viewed as a hypothesis class $\mathcal{H}_t$ (transfer architecture) consisting of all transfer networks with the induced architecture. This hypothesis class consists of multiclass hypotheses with instance space $\mathcal{X} = \mathbb{R}^{|V_0|}$ and output space $\mathcal{Y} = \{-1, 1\}^{|V_j|}$. $\mathcal{H}_u$ serves as the specific architecture. Their decomposition consists of the neural networks in $\mathcal{H}_{V,E,\sigma}$,

$$\mathcal{H}_u \circ \mathcal{H}_t = \{h_2 \circ h_1 \mid h_2 \in \mathcal{H}_u, h_1 \in \mathcal{H}_t\} = \mathcal{H}_{V,E,\sigma}$$

Each hypothesis class $B \in \mathcal{H}^I_{V,E,\sigma}$ is now treated as a neural network h_B with $M := |V_j|$ outputs and denote $h_B(\cdot)$ as its output.

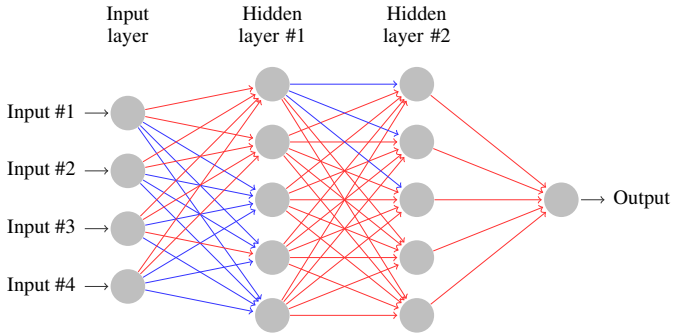

Figure 3: **A visualization of $\mathcal{H}^I_{V,E,\sigma}$:** I is the set of all the blue edges. Red edges are not transfered. Each bias $B_u \in \mathcal{H}^I_{V,E,\sigma}$ is determined by a fixed vector u consisting of the weights on the edges in I. Note that some activations are fed by both blue and red edges.

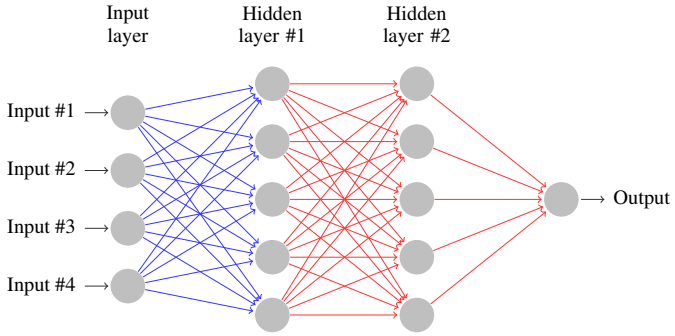

Figure 4: **A decomposition into transfer and specific networks:** the blue edges consist of the transfer network and the red ones are the specific network.

3 Problem setup

In Section 1 we introduced transfer learning as a multitask learning scenario with source tasks and target task. The learner is provided with data sets from similar (yet different) tasks and the goal is to come up with useful knowledge about the commonality between the tasks. That way, learning the target tasks would be easier, i.e., it would require smaller data sets.

In transfer learning, there are underlying and transfer problems. The underlying learning problem is the setting of each different learning problem. The transfer problem defines what is transferred during the learning process.

We follow the formalism of Baxter (2000) with some modifications. In our study, the underlying setting will be most of the time a realizable binary classification/regression setting with an instance set $Z = \mathcal{X} \times \mathcal{Y}$. The transfser setting $\mathcal{T} = (T, \mathscr{C}, \mathcal{E})$ is specified by,

- A hypothesis class family $\mathscr{C}$, which is a set of subsets of $\mathcal{H}$. With no loss of generality, we will assume that $\mathcal{H} = \bigcup_{B \in \mathscr{C}} B$.
- An environment $\mathcal{E}$, which is a set of tasks d.
- And an objective function $g(B, d) = \epsilon_d(B) := \inf_{c \in B} \epsilon_d(c)$. Typically, $B \in \mathscr{C}$.

The transfer learner has access to tasks $\{d_i\}_{i=1}^k \cup \{d_t\}$ (source and target) from $\mathcal{E}$. One approach to transfer learning is to come up with $B \in \mathscr{C}$ that fits these tasks well. The class B is called a *bias*. Learning a target task d_t might require fewer training examples, when B is learned successfully.

In traditional machine learning, data points are sampled i.i.d according to a fixed distribution. In transfer learning, samples are generated by what we call, a *Factory*. A factory is a process that provides multiple tasks. We suggest two major types of factories, *"Adversary Factories"* and *"'Ran-*

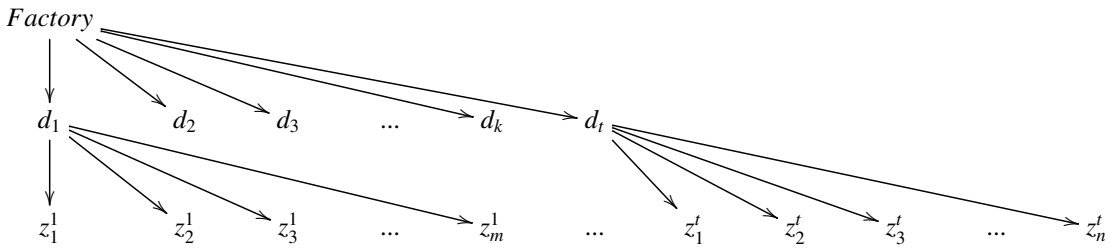

Figure 5: **A factory:** the sampling of samples z_j^i from tasks $\{d_i\}_{i=1}^k \cup \{d_t\}$. First, the tasks are selected either arbitrarily or randomly (depending on the factory type). The sample sets $s_i = \{z_j^i\}$ are then drawn from the corresponding tasks.

domized Factories". The first generates supervised tasks (i.e, distributions over $Z = \mathcal{X} \times \mathcal{Y}$). It selects concepts $\{c_i\}_{i=1}^k \cup \{c_t\}$ almost arbitrarily along to distributions over $\mathcal{X}$. The other selects the tasks randomly i.i.d from a distribution. In Section 4, we make use of adversary factories, while in Section 5 and Section 6 we use randomized factories instead.

In both cases, Figure 5 demonstrates the process done by the factory in order to sample training data sets. The difference between the two types arises from the method used to select the tasks.

3.1 The adversary factory

A factory selects k source tasks and a target task that the learner is tested on. An adversary factory is a type of factory that selects supervised tasks (i.e, distributions over $Z = \mathcal{X} \times \mathcal{Y}$). It selects source concepts $\{c_i\}_{i=1}^k$ that differ from the target concept c_t and are otherwise chosen arbitrarily. The factory also samples i.i.d distributions over $\mathcal{X}$, $\{p_i\}_{i=1}^k \cup \{p_t\}$, from the distribution of distributions $\mathcal{D}$. By the supervised behaviour of the learning setting, we have $\mathcal{E} = \mathcal{H} \times \mathcal{E}'$ where $\mathcal{E}'$ is a set of distributions over $\mathcal{X}$.

Definition 3 (Adversary factory)**.** *A factory $\mathcal{D}[k, m, n]$ is a process with parameters $[k, m, n]$ that:*

Step 1 *Selects $k+1$ tasks $d_1, ..., d_k, d_t$ such that $d_i = (p_i, c_i) \in \mathcal{E}$ in the following manner,*

- *Samples i.i.d $k+1$ distributions $p_1, p_2, .., p_k, p_t$ from a distribution of distributions $\mathcal{D}$.*
- *Selects any k source concepts $c_1, c_2, .., c_k$ and one target concept c_t out of $\mathcal{H}$ such that $\forall i \in [k] : \quad c_i \neq c_t$.*

Step 2 *Returns $S = (s_{[1,k]}, s_t)$ such that $s_i \sim d_i^m$ and $s_t \sim d_t^n$.*

Notation wise, if $n = 0$, we will write $\mathcal{D}[k, m]$ instead of $\mathcal{D}[k, m, 0]$. When $m = n = 0$, we will simply write $\mathcal{D}[k]$. To avoid symbol overload, similar notions will be used to denote randomized factories, depending on the section. For a data set S sampled according to an adversary factory, we denote with $O = (o_1, ..., o_k, o_t)$ the original data set without the labels. This data set is a sample according to $o_i \sim p_i^m$ and $o_t \sim p_t^n$ where $p_1, ..., p_k, p_t \sim \mathcal{D}^{k+1}$. In Section 4, all factories are adversary, while in the following sections they are randomized.

A K-factory is a factory that selects all concepts from $K \in \mathscr{C}$. K is said to be the subject of the factory. In this paper, we assume that all adversary factories are K-factories for some unknown bias $K \in \mathscr{C}$. The symbol K will be preserved to denote the subject a factory.

We will often write, in Section 4 and the associated proofs, $S \sim \mathcal{D}[k, m, n]$. This is a slight abuse of notation since the concepts are not samples. It would mean that the claim is true for any selection of the concepts. In some sense, we can assume that there is an underlying unknown arbitrary selection of concepts, and the data is sampled with respect to them. To avoid overload of notations, we will write $O \sim \mathcal{D}[k, m, n]$ to denote the corresponding unlabeled data set version of S.

The requirement that c_t differs from c_i for all $i \in [k]$ is essential to this first model. Intuitively, the interesting cases are those in which the target concept was not encountered in the source tasks.

Formally, it is easy to handle transfer learning using any learning algorithm, by just ignoring the source data. In the other direction, if we allow repeated use of the target concept, then any transfer algorithm can be used for conventional learning by repeatedly using the target data as the source. Thus, without the requirement, one cannot get meaningful transfer learning statements for adversary factories.

The knowledge that a set of tasks was selected from the same $K \in \mathscr{C}$, which is the subject of a K-factory, is the main source of knowledge made available during transfer learning. The second type of information arising from transfer learning is that all $k+1$ distributions were sampled from $\mathcal{D}$.

Using the face recognition example, there is the set of visual concepts $B_{i,1}$ that captures the appearances of different furniture, and there is the set of visual concepts $B_{i,2}$ that capture the characteristics of individual grasshoppers. From the source tasks, we infer that the target concept c_t belongs to a class of visual representations $B_{i,3}$ that contains image classifiers that are appropriate for modeling individual human faces.

For concreteness, we present our running example. A disk in $\mathbb{R}^3$ around 0 is a binary classifier that has radius r and a hyperplane h and is defined as follows:

$$f_{r,h}(x) = \begin{cases} 1 & \text{if } x \in h \cap B(r) \\ 0 & \text{if } o.w \end{cases}$$

Here, $B(r)$ is the ball of radius r around 0 in $\mathbb{R}^3$: $B(r) = \{x \in \mathbb{R}^3 : \|x\| \leq r\}$. We define $E_h = \{f_{r,h} : r \geq 0\}$, where, $\mathscr{C} = \{E_h : \forall h \text{ hyperplane in } \mathbb{R}^3 \text{ around } 0\}$. The following example demonstrates a specific K-factory on the hypothesis class defined above.

Example 1. *We consider a K-factory as follows: 1. the disks are selected arbitrarily with the same K such that the source concepts differ from the target concept, 2. D is supported with 3-D multivariate Gaussians with mean $\mu = 0$ and covariance $C = \sigma^2 I$, where σ is sampled uniformly in* $[1, 5]$.

Compatibility between biases In the adversary model, the source and target concepts are selected arbitrarily from a subject bias $K \in \mathscr{C}$. Therefore, we expect the ability of a bias B to approximate K to be the worst case approximation of a concept from K using a concept from B. Next, we formalize this relation that we call "compatibility".

We start with the mutual error rate, which is measured by $\epsilon_p(c_1, c_2) := \epsilon_d(c_1)$, where $d = (p, c_2)$. A bias B would be highly compatible with the factory's subject K if for every selection of a target concept from K, there is a good candidate in B. The compatibility of a bias B with respect to the subject K given an underlying distribution p over the instances set $\mathcal{X}$ is, therefore, defined as follows:

$$\textit{Compatibility Error Rate} = \mathrm{err}_p(B, K) := \sup_{c_2 \in K} \inf_{c_1 \in B} \epsilon_p(c_1, c_2)$$

In the adversary model, the distributions for the tasks are drawn from the same distribution of distributions. We, therefore, measure the generalization risk in the following manner:

$$\textit{Generalization Compatibility Error Rate} = E(B, K) = \mathbb{E}_{p \sim \mathcal{D}}\left[\mathrm{err}_p(B, K)\right]$$

The empirical counterparts of these definitions are given, for an unlabeled dataset o drawn from the distribution p as:

$$\textit{Empirical Compatibility Error Rate} = \mathrm{err}_o(B, K) = \sup_{c_2 \in K} \inf_{c_1 \in B} \epsilon_o(c_1, c_2)$$

where

$$\textit{Empirical Mutual Error Rate} = \epsilon_o(c_1, c_2) := \frac{1}{|o|} \sum_{x \in o} \ell(c_2, c_1(x))$$

In order to estimate $E(B, K)$, the average of multiple compatibility error rates is used. A set of unlabeled data sets $O = (o_1, ..., o_k)$ is introduced, each corresponding to a different source task, and the empirical compatibility error corresponding to the source data is measured by:

$$\textit{Source Empirical Compatibility Error Rate} = E_O(B, K) = \frac{1}{k} \sum_{i=1}^{k} \mathrm{err}_{o_i}(B, K)$$

3.2 The randomized factory

This randomized factory was presented, as matrix sampling, by Baxter (2000). In their learning to learn work transfer learning is not considered, and we modify the formulation to include the target task d_t.

Definition 4 (Randomized factory)**.** *A randomized factory (or simply, factory when the context is clear) is a process $\mathcal{D}[k, m, n]$ that:*

Step 1 *Samples i.i.d $k+1$ tasks $d_1, d_2, .., d_k, d_t \in \mathcal{E}$ from a distribution $\mathcal{D}$.*

Step 2 *Returns $S = (s_{[1,k]}, s_t)$ such that $s_i \sim d_i^m$ and $s_t \sim d_t^n$.*

The probabilistic nature of the the randomized factories allows them to fit a bias B by minimizing a suitable risk function. A natural choice of such a function is to measure the expected loss of B to approximate a task d with the following quantity, which we call *Transfer Generalization Risk*.

$$\epsilon_{\mathcal{D}}(B) := \mathbb{E}_{d\sim\mathcal{D}}[\epsilon_d(B)]$$

A reasonable approach to transfer learning is to first learn a bias B (in $\mathscr{C}$) that has a small transfer generalization risk. Since we typically have limited access to samples from the target task, we often employ the *Source Generalization Risk* instead:

$$\epsilon_U(B) := \frac{1}{k}\sum_{i=1}^{k}\epsilon_{d_i}(B), \quad \text{where} \quad U = \{d_1, ..., d_k\} \sim \mathcal{D}[k]$$

The definition of an adversary factory assumes that the concepts are selected arbitrarily, but without repetitions, from the same unknown hypothesis class $K \in \mathscr{C}$. The randomized factory that samples the concepts according to some distribution with the restriction of 0 probability for sampling the same concept twice, could be considered a special case. Our randomized factory results do not assume this 0 probability criteria. Nevertheless, this is the usual situation and the one that is of the most interest.

3.3 Transferability

In this section, we provide general definitions of transfer learning. We follow the classical learning theory: defining a PAC notion of transfer learning and VC-like dimensions. We then introduce a family of learning rules applicable for transfer learning. After describing the theory, we will turn to proving a fundamental Theorem that states these are all equivalent to PAC-learnability.

Definition 5 (PAC-transfer)**.** *A transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ is PAC-transferable if:*

$$\exists \text{ algorithm } A \quad \forall \epsilon, \delta \quad \exists k_0, m_0, n_0 \text{ (functions of } \epsilon, \delta)$$
$$\forall k > k_0, \ m > m_0, \ n > n_0 \quad \forall \mathcal{D}:$$
$$\mathbb{P}_{S\sim\mathcal{D}[k,m,n]}\left[\epsilon_{d_t}(A(S)) \leq \inf_{c\in\mathcal{H}}\epsilon_{d_t}(c) + \epsilon\right] \geq 1-\delta$$

where $A(S) \in \mathcal{H}$ is the output of the algorithm, and (k_0, m_0, n_0) are three functions of (ϵ, δ).

This model relies on the original PAC-learning model. In the classical PAC model, a learning algorithm is introduced. The algorithm samples enough labeled data examples from an arbitrary distribution, labeled by a target concept. The output is a hypothesis that has a high probability of classifying correctly a new example (small error on the target task), with high confidence. In our framework, the idea is similar. In this case, the learner has access to examples from different tasks. The learner's hope is to be able to come up with useful common knowledge, from the source tasks, for learning a new concept for the target task. The output is a hypothesis that has a small error on the target task. In this case, the factory (chosen arbitrarily) provides the data samples. The main assumption is that the distributions from which the examples are selected are sampled i.i.d from the same distribution of distributions. In many cases, we will provide a realizable assumption that all concepts share the same representation (i.e, in the same $K \in \mathscr{C}$). As we already mentioned, this is the case when dealing with adversary factories. It provides common knowledge between the concepts. The probabilistic assumption enables the algorithm to transfer that useful information.

Next, we define VC-like dimensions for factory-based transfer learning. Unlike conventional VC dimensions, which are purely combinatorial, the suggested dimensions are algorithmic and probabilistic. This is because the post-transfer learning problem relies on information gained from the source samples.

Definition 6 (Transfer VC dimension)**.** *$\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ has transfer VC dimension $\leq vc$ if:*

$$\exists \text{ algorithm } N \;\; \forall \epsilon, \delta \;\; \exists k_0, m_0 \text{ (functions of } \epsilon, \delta) \; \forall k > k_0, \; m > m_0 \;\; \forall \mathcal{D} :$$
$$\mathbb{P}_{S \sim \mathcal{D}[k,m]} \left[vc(N(S)) \leq vc \text{ and } \inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \inf_{c \in \mathcal{H}} \epsilon_{d_t}(c) + \epsilon \right] \geq 1 - \delta$$

Here, $N(S) \in \mathscr{C}$ is a hypothesis class. We say that the transfer VC dimension is exactly d, if the above expression does not hold with d replaced with $vc - 1$.

The algorithm N is called *narrowing*. These algorithms are special examples of how the common knowledge might be extracted. In the first stage the algorithm, that is provided with source data returns a narrow hypothesis class $N(S)$ that with a high probability approximates very well on d_t. $N(S)$ can be viewed as a learned representation of the tasks from $\mathcal{D}$. Post-transfer, learning takes place in $N(S)$, where there exists a hypothesis that is ϵ-close to the best approximation possible in $\mathcal{H}$. In different situations, we will assume realizability, i.e, there exists $B \in \mathscr{C}$ such that $\mathcal{D}$ is supported by tasks d that satisfy $\inf_{c \in B} \epsilon_d(c) = 0$. In the adversary case, each factory is a K-factory and, in particular, realizable.

In the face recognition example, the deep learning algorithm has access to face images of multiple humans. From this set of images, a representation of an image of human faces is learned. Next, given the representation of human faces, the learner selects a concept that best fits the target data in order to learn the specified human face.

By virtue of general VC theory, for learning a target task, with enough target examples (w.r.t. the capacity of $N(S)$ instead of $\mathcal{H}$'s capacity), one is able to output a hypothesis that is $(\epsilon + \epsilon')$-close to the best approximation in $\mathcal{H}$, where ϵ' is the accuracy parameter of the post-transfer learning algorithm.

We, therefore, define a 2-step program. The first step applies narrowing and replaces $\mathcal{H}$ by a simplified hypothesis class $B = N(s_{[1,k]})$. The second step learns the target concept within B. An immediate special case, the T-ERM learning rule (transfer empirical risk minimization), uses an ERM rule as its second step. Put differently,

Input: $S = (s_{[1,k]}, s_t)$.
Output: concept c_{out} such that $\epsilon_{d_t}(c_{out}) \leq \epsilon$ with probability $\geq 1 - \delta$.

Narrowing narrow the hypothesis class $\mathcal{H} \mapsto B := N(s_{[1,k]})$;

Output $c_{out} = \mathrm{ERM}_B(s_t)$;

Algorithm 1: T-ERM learning rule

In the following sections, when needed, we will use the following to denote the minimal target sample complexity of 2-step programs with n_{2step} (a function of ϵ, δ).

We claim that whenever T is a learnable binary classification learning setting, once the narrowing is performed, the ERM step is possible, with a number of samples that depend only on ϵ, δ. For this purpose, the following Lemma is useful:

Lemma 1. *The sample complexity of any learnable binary classification hypothesis class $\mathcal{H}$ of VC dimension u, is bounded by a universal function $N_u(\epsilon, \delta)$. i.e, it depends only on the VC dimension.*

Proof. Simply by Theorem 1.

Based on Lemma 1, with enough k, m (sufficient to apply the narrowing with error and confidence parameters $\epsilon/2, \delta/2$) and $n = N_u(\epsilon/2, \delta/2)$ from Lemma 1, the T-ERM rule returns a concept that has error $\leq \epsilon$ with probability $\geq 1 - \delta$.

4 Results in the adversary model

In this section, we make use of the adversary factories in order to present the equivalence between the different definitions of transferability discussed above for the binary case. Furthermore, it is shown that in this case, transferability is equivalent to PAC-learnability. In the next section, we study the advantages of transfer learning that exist despite this equivalence.

4.1 Transferability vs. learnability

The binary classification case

Theorem 3. *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a binary classification transfer learning setting. The following conditions on $\mathscr{T}$ are then equivalent:*

1. *Has finite transfer dimension.*
2. *Is PAC-transferable.*
3. *Is PAC-learnable.*

Next, we provide bounds on the target sample complexity of 2-step programs.

Corollary 1 (Quantitative results)**.** *When $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ is a binary classification transfer learning setting that has transfer VC dimension vc, then the following holds:*

$$C_1 \cdot \left(\frac{vc + \log(1/\delta)}{\epsilon} \right) \leq n_{2step}(\epsilon, \delta) \leq C_2 \cdot \left(\frac{vc + \log(1/\delta)}{\epsilon^2} \right)$$

For some constants $C_1, C_2 > 0$.

Proof. This corollary follows immediately from the characterization above and is based on Blumer et al. (1989). We apply narrowing in order to narrow the class to VC dimension $\leq v$. The second step learns the hypothesis in the narrow subclass. The upper bound follows when the narrow subclass differs from K (unrealizable case) while the lower bound turns in when K equals the narrow subclass (realizable case). □

The regression case We demonstrate that transferability does not imply PAC-learnability in regression problems. PAC-learning and PAC-transferability are well defined for regression using appropriate losses. As the following Lemma shows, in the regression case, there is no simple equivalence between PAC-transferability and learnability.

Lemma 2. *There is a transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ that is PAC-transferable but not PAC-learnable with squared loss ℓ.*

While the example in the proof of Lemma 2 (Appendix) is seemingly pathological, the scenario of non-learnability in regression is common, for example, due to colinearity that gives rise to ill-conditioned learning problems. Having the ability to learn from source tasks reduces the ambiguity.

4.2 Trivial and non-trivial transfer learning

Transfer learning would be beneficial if it reduces the required target sample complexities. We call this the "non-trivial transfer" property. It can also be said that a transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ is non-trivial transferable, if there is a transfer learning algorithm for it with a target sample complexity smaller than the sample complexity of any learning algorithm of $\mathcal{H}$ by a factor $0 < c < 1$. An alternative definition (that is not equivalent) is saying that the VC transfer and regular VC dimensions differ. We next describe a pathological case in which transfer is trivial, and then demonstrate the existence of non-trivial transfer.

The pathological case can be demonstrated in the following simple example. Let $\mathcal{H}$ be the set of all 2D disks in $\mathbb{R}^3$ around 0. Each E_h contains the disks on the same hyperplane h, for a finite collection of h. Consider the factory $\mathcal{D}$ that samples distributions d supported only by points from

the hyperplanes h with a distance of at least 1 from the origin. Since, in our model, the concepts are selected arbitrarily, consider the case where all source concepts are disks with a radius smaller than 1, and the target concept has a radius of 2. In the source data, all examples are negative, and no information is gained on the hyperplane h.

Despite the existence of the pathological case above, the following Lemma claims the existence of non-trivial transferability.

Lemma 3. *There exists a binary classification transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ (i.e, T is a binary classification setting) that is non-trivial transferable.*

4.3 Generalization bounds for adversary transfer learning

In the previous section, we investigated the relationship between learnability and transferability. It was demonstrated that, in some cases, there is non-trivial transferability. In such cases, transfer learning is beneficial and helps to reduce the size of the target data.

In this section, we extend the discussion on non-trivial transferability. We focus on generalization bounds for transfer learning in the adversary model. Two bounds are presented. The first bound is a VC-style bound. The second bound combines both PAC-Bayesian and VC perspectives. The proposed bounds will shed some light about representation learning and transfer learning in general. Nevertheless, despite the wide applicability of these generalization bounds, it will not be trivial to derive a transfer learning algorithm from them since they measure the difference between generalization and empirical compatibility error rates. In general, computing the empirical compatibility error rate requires knowledge about the subject of the factory, which is kept unknown. Therefore, without additional assumptions it is intractable to compute this quantity.

VC-style bounds for adversary transfer learning We extend the original VC generalization bound for the case of adversary transfer learning. We call the presented bound, "The min-max transfer learning bound". In this context, the min-max stands for the competition between the difficulty to approximate K and the ability of a bias B to approximate it.

This bound estimates the expected worst case difference between the generalization compatibility of B to K and the empirical source compatiblity of B and K. The upper bound is the sum of two regularization terms. The first penalizes both complexities of B and K with respect to the number of samples per task, m. The second penalizes on the complexity of $\mathscr{C}$ with respect to k.

The first step towards constructing a VC bound in the adversary model is defining a growth function specialized for this setting. The motivation is controlling the compatiblity between a bias B and the subject K. Throughout the construction of compatibility measurements, the most elementary unit is the empirical error of B, $\min_{c_1 \in B} \epsilon_o(c_1, c_2)$ for some $c_2 \in K$ along to an unlabeled data set o. Instead of dealing with the whole bias B, we can focus only on $c_1 = \mathrm{ERM}_B(c_2(o))$. In that way we can control the compatibility of B with K on the data set o. In transfer learning, we wish to control the joint error. Put differently, in the average of multiple compatibility errors on different data sets. For this purpose, we count the number of different configurations of two concepts $c_{i,1}$ and $c_{i,2}$ on unlabeled data sets o_i such that $c_{i,1} = \mathrm{ERM}_B(c_{i,2}(o_i))$.

To avoid notaional overload, we assume that the ERM is fixed, i.e., we assume an inner implementation of an ERM rule that takes a data set and returns a hypothesis for any selected bias B. Nevertheless, we do not restrict how it is implemented. More formally, $\mathrm{ERM}_B(s)$ represents a specific function that takes B, s and returns a hypothesis in B.

Based on this background, we denote the following set of configurations:

$$[\mathcal{H}, \mathscr{C}, K]_O = \{(c_{1,1}(o_1), c_{1,2}(o_1), ..., c_{k,1}(o_k), c_{k,2}(o_k)) : c_{i,2} \in K \text{ and } c_{i,1} = \mathrm{ERM}_B(c_{i,2}(o_i)) \text{ s.t } B \in \mathscr{C}\}$$

In addition, the *Adversarial Transfer Growth Function* $\tau(k, m; \mathscr{C}, K)$,

$$\tau(k, m; \mathscr{C}, K) = \max_{O \in \mathcal{X}^{mk}} \left| [\mathcal{H}, \mathscr{C}, K]_O \right|$$

This quantity represents the worst case number of optional configurations.

Theorem 4 (The min-max transfer learning bound)**.** *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a binary classification transfer learning setting. Then,*

$$\forall \mathcal{D} \;\; \forall K \in \mathscr{C} : \mathbb{E}_{O\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|E(B,K) - E_O(B,K)\right|\right] \\ \leq \frac{4 + \sqrt{\log(\tau(2k, m; \mathscr{C}, K))}}{\sqrt{2k}} + \frac{4 + \sqrt{\log(\sup_B \tau_B(2m)) + \log(\tau_K(2m))}}{\sqrt{2m}}$$

PAC-Bayes bounds for adversary transfer learning This bound combines between PAC-Bayesian and VC perspectives. We call it "The perturbed min-max transfer learning bound". This is because there is still a competition between the ability of the bias to approximate and the difficulty of the subject. Nevertheless, in this case the bias is perturbed.

We take a statistical relaxation of the standard model. A set of posterior distributions $\mathcal{Q}$ and a prior distribution P, both over $\mathscr{C}$ are taken. Extending the discussion in Section 2, the aim is being able to select $Q \in \mathcal{Q}$ that best fit the data instead of a concrete bias. In this setting, we measure an expected version of the generalization compatibility error rate with B distributed by $Q \in \mathcal{Q}$. We call it the *Expected Generalization Compatibility Error Rate*. Formally,

$$E(Q, K) = \mathbb{E}_{B\sim Q}\mathbb{E}_{p\sim\mathcal{D}}\left[\mathrm{err}_p(B, K)\right]$$

It is important to note that the left hand side of the bound is, in general, intractable to compute. This is due to its direct dependence on K which is unknown. Nevertheless, there still might be conditions in which different learning methods do minimize this argument. In addition, it gives insights on what a "good" bias is.

Theorem 5 (The perturbed min-max transfer learning bound)**.** *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a binary classification transfer learning setting. In addition, P a prior distribution and $\mathcal{Q}$ a family of posterior distributions, both over $\mathscr{C}$. Let $\delta \in (0, 1)$ and $\lambda > 0$, then for all factories $\mathcal{D}$ with probability $\geq 1 - \delta$ over the selection of $O \sim \mathcal{D}[k, m]$,*

$$\forall Q \in \mathcal{Q}, K \in \mathscr{C} : E(Q, K) \leq \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}[\mathrm{err}_{o_i}(B, K)] \\ + \sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m)) + \log(8/\lambda\delta)}{m}} + \frac{1}{m} + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}} + \lambda\delta$$

With the restriction that $k \geq \frac{8\log\left(\frac{2}{\delta}\right)}{(\lambda\delta)^2}$.

5 Results in the randomized model

We start our discussion of randomized factories with the following Lemma that revisits the disks in $\mathbb{R}^3$ example above for this case.

Lemma 4. *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a realizable transfer learning setting such that $\mathcal{H}$ is the set of all 2D disks in $\mathbb{R}^3$ around 0. Each E_h = disks on the same hyperplane h and $\mathscr{C} = \{E_h : \forall h$ hyperplane in $\mathbb{R}^3$ around $0\}$. This hypothesis class has transfer VC dimension = 1 (and regular VC dimension = 2).*

5.1 Transferability vs. learnability

A transferring rule or bias learner N is a function that maps a source data (i.e, $s_{[1,k]}$) into a bias B. An interesting special case is the *simplifier*. A simplifier fits $B \in \mathscr{C}$ that has a relatively small error rate.

Definition 7 (Simplifier)**.** *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a transfer learning setting. An algorithm N with access to $\mathscr{C}$ is called a simplifier if:*

$$\forall \epsilon, \delta \;\; \exists k_0, m_0 \text{ (functions of } \epsilon, \delta) \; \forall k > k_0, \; m > m_0 \;\; \forall \mathcal{D} : \\ \mathbb{P}_S\left[\epsilon_{\mathcal{D}}(N(S)) \leq \inf_{B\in\mathscr{C}} \epsilon_{\mathcal{D}}(B) + \epsilon\right] \geq 1 - \delta$$

Here, the source data S is sampled according to $\mathcal{D}[k,m]$. In addition, $N(S) \in \mathscr{C}$, which is a hypothesis class, is the result of applying the algorithm to the source data. The quantities k_0, m_0 are functions of ϵ, δ.

The standard ERM rule is next extended into a transferring rule. This rule returns a bias B that has the minimum error rate on the data, measured for each data set separately. This transferring rule, called $\mathrm{C-ERM}_{\mathscr{C}}(S)$ is defined as follows,

$$\boxed{\mathrm{C-ERM}_{\mathscr{C}}(S) := \arg\min_{B\in\mathscr{C}} \frac{1}{k}\sum_{i=1}^{k} \epsilon_{s_i}(c^*_{i,B}),\ \text{s.t}\ c^*_{i,B} = \mathrm{ERM}_B(s_i)}$$

This transferring rule was previously considered by Ando et al. (2005) who named it Joint ERM.

Uniform convergence (Shalev-shwartz et al. (2010)) is defined for every hypothesis class $\mathcal{H}$ (w.r.t loss ℓ), in the usual manner:

$$\forall d : \mathbb{P}_{S\sim d^k}\left[\forall c \in \mathcal{H} : \left|\epsilon_d(c) - \epsilon_s(c)\right| \leq \epsilon\right] \geq 1-\delta$$

It can also be defined for $\mathscr{C}$ (w.r.t loss g) as:

$$\forall \mathcal{D} : \mathbb{P}_{U\sim\mathcal{D}[k]}\left[\forall B \in \mathscr{C} : \left|\epsilon_{\mathcal{D}}(B) - \epsilon_U(B)\right| \leq \epsilon\right] \geq 1-\delta$$

For any k larger than some function $k(\epsilon,\delta)$.

The following lemma states that whenever both $\mathcal{H}$ and $\mathscr{C}$ have uniform convergence properties, then the $\mathrm{C-ERM}_{\mathscr{C}}$ transferring rule is a simplifier for $\mathcal{H}$.

Lemma 5. *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a transfer learning setting. If both $\mathscr{C}$ and $\mathcal{H}$ have uniform convergence properties, then the* $\mathrm{C-ERM}$ *rule is a simplifier of $\mathcal{H}$.*

The preceeding Lemma explained that $\mathrm{C-ERM}_{\mathscr{C}}$ transferring rules are helpful for transferring knowledge efficiently.

The next Theorem states that even if the hypothesis class $\mathcal{H}$ has an infinite VC dimension, there still might be a simplifier outputing hypothesis classes with a finite VC dimension. This result, however, could not be obtained when restricting the size of the data to be bounded by some function of ϵ, δ. Therefore, in a sense, there is transferability beyond learnability.

Theorem 6. *The following statements hold on binary classification.*

- *There is a binary classification transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ such that $\mathcal{H}$ has an infinite VC dimension, has a simplifier N that always outputs a finite VC dimensional bias B.*
- *If a binary classification transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ has an infinite VC dimension, then $\sup_{B\in\mathscr{C}} vc(B) = \infty$.*

5.2 Generalization bounds for randomized transfer learning

In the previous section, we explained that whenever both $\mathcal{H}$ and $\mathscr{C}$ have uniform convergence properties, there exists a simplifier for this transfer learning setting. We explained that, in this case, a C-ERM rule is an appropriate simplifier. In this section, we widen the discussion on the existence of a simplifier for different cases. For this purpose, we extend famous generalization bounds from statistical learning theory to the case of transfer learning.

VC-style bounds for transfer learning We begin with an extension of the original VC bound (see Section 2) to the case of transfer learning. It upper bounds the expected (w.r.t random source data) difference between the transfer generalization risk of B and the *2-step Source Empirical Risk* working on B. A mapping $r : B \mapsto r_B$ from a bias B to a learning rule of the bias (i.e, outputs hypotheses in B with empirical error that converge to $\inf_{c\in B} \epsilon_d(c)$) is called *post transfer learning*

rule/algorithm (see Equation 2). Informally, the 2-step source empirical risk measures the empirical success rate of a post transfer learning rule on a few data sets. The bounding quantity depends on the ability of the post transfer learning rule to generalize.

The 2-step source empirical risk is formally defined as follows,

$$\epsilon_S(B,r) := \frac{1}{k}\sum_{i=1}^{k}\epsilon_{s_i}(r_B(s_i)),\ \text{s.t}\ \ S = s_{[1,k]}$$

Next, the standard constructions of the original VC bound are extended,

$$[\mathcal{H},\mathscr{C},r]_S = \{r_{1,B}(s_1),...,r_{k,B}(s_k) : B \in \mathscr{C}\},\ \text{s.t}\ \ S = s_{[1,k]}$$

Here, $r_{i,B} := r_B(s_i)$ denotes the application of the learning rule r_B on s_i and $r_{i,B}(s_i)$, the realization of $r_{i,B}$ on s_i. The equivalent of the standard growth function in transfer learning is the *transfer growth function*,

$$\tau(k,m,r) = \max_S \left|[\mathcal{H},\mathscr{C},r]_S\right|$$

With this formalism, we can state our extended version of the VC bound.

Theorem 7 (Transfer learning bound 1)**.** *Let $\mathscr{T} = (T,\mathscr{C},\mathcal{E})$ be a binary classification transfer learning setting such that T is learnable. In addition, assume that r is a post transfer learning rule, i.e, endowed with the following property,*

$$\forall d, B:\ \ r_B(\cdot) \in B \text{ and } \mathbb{E}_{s\sim d^m}\left[\left|\inf_{c\in B}\epsilon_d(c) - \epsilon_s(r_B(s))\right|\right] \le \epsilon(m) \to 0 \tag{2}$$

Then,

$$\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{\mathcal{D}}(B) - \epsilon_S(B,r)\right|\right] \le \frac{4+\sqrt{\log(\tau(2k,m,r))}}{\sqrt{2k}} + \epsilon(m)$$

We conclude that in binary classification, if T is learnable, and r satisfies Equation 2 then, there exists a simplifier whenever the transfer growth function $\tau(k,m,r)$ is polynomial in k.

PAC-Bayes bounds for transfer learning We provide two different PAC-Bayes bounds for transfer learning. The first bound estimates the gap between the generalization transfer risk of each $B \in \mathscr{C}$ and the average of the empirical risks of $c^*_{i,B}$ in the binary classification case. On the other hand, Theorem 9 will argue a more general case when $\mathcal{H}$ might have an infinite VC dimension or the underlying learning setting is not binary classification.

The first approach concentrates on model selection within PAC-Bayesian bounds. It presents a bound for model selection that combines PAC-Bayes and VC bounds. We construct a generalization bound to measure the fitting of a random representation. i.e, the motivation is searching for Q that minimizes,

$$R(Q) = \mathbb{E}_{B\sim Q}\mathbb{E}_{d\sim\mathcal{D}}\left[\inf_{c\in B}\epsilon_d(c)\right]$$

Theorem 8 (Transfer learning bound 2)**.** *Let $\mathscr{T} = (T,\mathscr{C},\mathcal{E})$ be a binary classification transfer learning setting. In addition, P a prior distribution and $\mathcal{Q}$ a family of posterior distributions, both over $\mathscr{C}$. Let $\delta \in (0,1)$ and $\lambda > 0$, then with probability $\ge 1-\delta$ over S,*

$$\forall Q \in \mathcal{Q} : R(Q) \le \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}\left[\epsilon_{s_i}(c^*_{i,B})\right]$$
$$+\sqrt{\frac{\log(\tau_{\mathcal{H}}(2m)) + \log(8/\lambda\delta)}{m}} + \frac{1}{m} + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}} + \lambda\delta$$

With the restriction that $k \ge \frac{8\log(2/\delta)}{(\lambda\delta)^2}$.

We then derive the following randomized transferring rule,

$$\boxed{\operatorname{argmin}_Q \left[\frac{1}{k} \sum_{i=1}^{k} \mathbb{E}_{B\sim Q} \left[\epsilon_{s_i}(c^*_{i,B}) \right] + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}} \right]}$$

Which is helpful only when both k, m tend to increase.

The previous bound relied on the assumption that the learning setting is binary classification and the underlying learning setting is learnable. Next, we suggest a different approach to PAC-Bayes bounds for transfer learning. The current bound is pure PAC-Bayesian and is more related to Pentina & Lampert (2014). In their work, the motivation is to be able to learn a prior distribution for learning new tasks. The aim is measuring the effectiveness of a prior distribution for learning new tasks with a selected learning rule. The weakness in their analysis is that it relies on the assumption that the source training data sets and the target training data set are i.i.d distributed and thus proportional in their sizes. In this work, we suggest a different perspective for PAC-Bayes transfer bounds that overcomes this problem.

The first step towards the construction of the bound is to adopt a generalized PAC-Bayesian setting.

- A transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$.
- P a prior distribution and $\mathcal{Q}$ a family of posterior distributions, both over $\mathscr{C}$.
- p a prior distribution and

$$\mathcal{U} = \left\{ Q_q(c) = \int_B Q(B) \cdot q(c; B)\, dB : Q \in \mathcal{Q}, q \right\}$$

  a family of posterior distributions, both over $\mathcal{H}$.

The set $\mathcal{U}$ consists of all distributions that first sample a subset B from Q and then sample c from $q(\cdot; B)$ that is over B. We are again interested in finding Q that minimizes,

$$R(Q) = \mathbb{E}_{B\sim Q} \mathbb{E}_{d\sim\mathcal{D}} \left[\inf_{c\in B} \epsilon_d(c) \right]$$

Theorem 9 (Transfer learning bound 3)**.** *Assume the PAC-Bayesian framework above. Let $\delta \in (0, 1)$ and $\lambda > 0$, then with probability $\geq 1 - \delta$ over $S \sim \mathcal{D}[k, m]$, the following holds for all $Q \in \mathcal{Q}$,*

$$R(Q) \leq \frac{1}{k} \sum_{i=1}^{k} \min_{q_i} \mathbb{E}_{c\sim Q_{q_i}} \left[\epsilon_{s_i}(c) \right] + \sqrt{\frac{\mathrm{KL}(Q_{q_i}\|p) + \log(2m/\lambda\delta)}{2(m-1)}} + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}} + \lambda\delta$$

With the restriction that $k \geq \frac{8\log(2/\delta)}{(\lambda\delta)^2}$.

As for the previous bound, we can arrive to a different transferring rule,

$$\boxed{\operatorname{argmin}_Q \left[\frac{1}{k} \sum_{i=1}^{k} \left(\min_{q_i} \mathbb{E}_{c\sim Q_{q_i}} \left[\epsilon_{s_i}(c) \right] + \sqrt{\frac{\mathrm{KL}(Q_{q_i}\|p) + \log(2m/\lambda\delta)}{2(m-1)}} \right) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}} \right]}$$

6 Deep transfer learning

6.1 VC-style bounds for deep transfer learning

It is interesting to show how Theorem 7 can be applied to the case of deep learning. We provide a VC-like bound for deep learning. This will be a major step towards proving nontrivial transferability for a very wide class of neural network architectures. In addition, it will give insights on major open

questions like "deep architectures vs. shallow architectures", "expressivity of deep architectures" and "generalization ability of deep architectures" in their general aspect and in the particular case of transfer learning.

We study the case when the architecture decomposes into transfer and specific architectures $\mathcal{H}_t$ and $\mathcal{H}_u$ (see Section 2). For each bias B, we denote its corresponding neural network with h_B.

First, we show that the growth function of the transfer learning setting can be bounded with the growth function of $\mathcal{H}_t$. Denote $\tau_t(\cdot)$ the growth function of the hypothesis class $\mathcal{H}_t$.

We assume that the produced labels of a post transfer learning rule r are independent of B and s given $h_B(s)$. i.e,

$$r_{B_1}(s_1)(s_1) = r_{B_2}(s_2)(s_2) \text{ whenever } h_{B_1}(s_1) = h_{B_2}(s_2) \tag{3}$$

It can also be stated that, if $r_{B_1}(s_1)$ and $r_{B_2}(s_2)$ are trained hypotheses under the assumption that $h_{B_1}(s_1) = h_{B_2}(s_2)$, then their labelings on s_1 and s_2 are the same. We will next show that this assumption is common when the hypothesis class can be decomposed to $\mathcal{H}_u \circ \mathcal{H}_t$.

Lemma 6. *Let $\mathscr{T} = (T, \mathcal{H}^I_{V,E,sign}, \mathcal{E})$ be a transfer learning setting such that $T = (\mathcal{H}_{V,E,sign}, Z, \ell)$ and ℓ is the 0-1 loss. Assume that I consists of all edges between the first j layers. Let r be any post transfer learning rule (i.e, a mapping $r : B \to r_B$ such that $r_B(s) \in B$ for all finite $s \subset Z$) satisfying 3. Then,*

$$\tau(k, m, r) \leq \tau_t(mk) \leq (mke)^{|I|}$$

Where τ_t is the growth function of the hypothesis class $\mathcal{H}_t$.

Plug in Lemma 6 into Theorem 7 for the proposed deep learning setting and arrive at the following generalization bound.

Theorem 10 (Deep learning bound 1)**.** *Let $\mathscr{T} = (T, \mathcal{H}^I_{V,E,sign}, \mathcal{E})$ be a transfer learning setting such that $T = (\mathcal{H}_{V,E,sign}, Z, \ell)$ and ℓ is the 0-1 loss. In addition, assume that r satisfies Equation 3 and Equation 2. Then,*

$$\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{\mathcal{D}}(B) - \epsilon_S(B, r)\right|\right] \leq \frac{4 + \sqrt{|I|\log(mke)}}{\sqrt{2k}} + \epsilon(m)$$

Proof. An application of Theorem 7 for the discussed case with the bound from Lemma 6.

Theorem 11 (Deep learning bound 2)**.** *Let $\mathscr{T} = (T, \mathcal{H}^I_{V,E,sign}, \mathcal{E})$ be a transfer learning setting such that $T = (\mathcal{H}_{V,E,sign}, Z, \ell)$, ℓ is the 0-1 loss and denote $E = I \cup J$ (where $I \cap J = \emptyset$). In addition, assume that $r : B \to \mathrm{ERM}_{\mathcal{H}_u}(h_B(\cdot)) \circ h_B$. Then,*

$$\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{\mathcal{D}}(B) - \epsilon_S(B, r)\right|\right] \leq \frac{4 + \sqrt{|I|\log(mke)}}{\sqrt{2k}} + \frac{8 + \sqrt{4|J|\log(2me)}}{\sqrt{2m}}$$

Where $r_B := \mathrm{ERM}_{\mathcal{H}_u}(h_B(\cdot)) \circ h_B$ is a learning rule that takes a data set s and outputs $\mathrm{ERM}_{\mathcal{H}_u}(h_B(s)) \circ h_B$ in return.

This Theorem asserts that nontrivial transferability holds among a very general class of transfer learning settings of neural networks. As we have shown, whenever the architecture is divided into transfer and specific parts, a narrowing process reduces the whole hypothesis class of neural networks into one that has lower capacity. In a realizable case (i.e, there is B such that for all d we have $\inf_{c\in B} \epsilon_d(c) = 0$), the transfer VC dimension might decrease.

6.2 PAC-Bayes bounds in deep transfer learning

We next apply the PAC-Bayesian bounds of Section 5.2 to the case of neural networks.

The motivation is transferring common weights between neural networks. It is preferable to use $\mathcal{H}^I_{V,E,\sigma}$ as the hypothesis class family. Inspired by the Gaussian parameterization for neural networks presented by McAllester (2013), a bias corresponding to weight vector u is identified with a Gaussian distribution centered by u. Formally,

$$Q_u \sim N(u, \mathbb{1}),\ P \sim N(0_{|I|}, \mathbb{1}) \text{ and } \mathcal{Q} = \left\{Q_u \mid u \in \mathbb{R}^{|I|}\right\} \implies \mathrm{KL}(Q_u \| P) = \|u\|^2/2 \tag{4}$$

Where $0_{|I|}$ is a ($|I|$-dimensional) vector of zeros and $\mathbb{1}$ is a unit matrix (of dimension $|I| \times |I|$).

Theorem 12 (Deep learning bound 3)**.** *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a transfer learning setting such that $\mathcal{H} := \mathcal{H}_{V,E,sign}$ and $\mathscr{C} := \mathcal{H}^I_{V,E,sign}$ for any neural network architecture (V, E, σ) and $I \subset E$. P and $\mathcal{Q}$ as above. Let $\delta \in (0,1)$ and $\epsilon > 0$, then with probability $\geq 1 - \delta$ over S, for all u,*

$$R(Q_u) \leq \frac{1}{k}\sum_{i=1}^{k} \mathbb{E}_{B\sim Q_u}\left[\epsilon_{s_i}(c^*_{i,B})\right] + \sqrt{\frac{|E|\log(2me) + \log(8/\epsilon)}{m}} + \frac{1}{m} + \sqrt{\frac{\|u\|^2/2 + \log(2k/\delta)}{2(k-1)}} + \epsilon$$

With the restrictions that $k \geq \frac{8\log(2/\delta)}{\epsilon^2}$.

Proof. An application of Theorem 8 with the explanations above and $\lambda\delta = \epsilon$.

Post transfer (after selecting $B := B^*$ that best fits the data), one is able to use the common knowledge extracted in the transfer step in order to learn a new concept. One approach to do it is by fixing I'th weights to be B^*'s and learning only the rest of the weights. Formally, we learn the target task within the hypothesis class B^* that consists of all neural networks with architecture (V, E, sign) and I's weights are B^*'s vector.

Lemma 7. *The VC dimension of each $B \in \mathcal{H}^I_{V,E,sign}$ is $vc(B) = O\left(|J|\log|J|\right)$.*

Following the same line, we apply Theorem 9 with Gaussian distributions. As before, each bias B_u is identified with a Gaussian distribution centered by u. In addition, a neural network $h_{V,E,\text{sign},w} \in B_u$, is identified with the weight vector $w = u\|v$ (concatenation of two vectors) consisting of all the weights on E. We fit a Gaussian distribution centered by this weights vector to parameterize $h_{V,E,\text{sign},w}$. We select,

$$\mathcal{Q} = \left\{Q_u \sim N(u, \mathbb{1}) : u \in \mathbb{R}^{|I|}\right\},\ P \sim N(0_{|I|}, \mathbb{1})$$
$$\implies \mathrm{KL}(Q_u\|P) = \|u\|^2/2$$

And,

$$\mathcal{U} = \left\{Q_{u,v} \sim N(u,\mathbb{1})\cdot N(v,\mathbb{1}) : u \in \mathbb{R}^{|I|}, v \in \mathbb{R}^{|J|}\right\},\ p \sim N(0_{|E|}, \mathbb{1})$$
$$\implies \mathrm{KL}(Q_{u,v}\|p) = \|u\|^2/2 + \|v\|^2/2$$

Theorem 13 (Deep learning bound 4)**.** *Let $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$ be a transfer learning setting such that $\mathcal{H} := \mathcal{H}_{V,E,sign}$ and $\mathscr{C} := \mathcal{H}^I_{V,E,sign}$ for any neural network architecture (V, E, σ) and $I \subset E$. P, $\mathcal{Q}$, p and $\mathcal{U}$ as above. Let $\delta \in (0,1)$ and $\epsilon > 0$, then with probability $\geq 1 - \delta$ over S, for all u,*

$$R(Q_u) \leq \frac{1}{k}\sum_{i=1}^{k} \min_{v_i} \mathbb{E}_{c\sim Q_{u,v_i}}\left[\epsilon_{s_i}(c)\right] + \sqrt{\frac{\|u\|^2/2 + \|v_i\|^2/2 + \log(2m/\epsilon)}{2(m-1)}} + \sqrt{\frac{\|u\|^2/2 + \log(2k/\delta)}{2(k-1)}} + \epsilon$$

With the restriction that $k \geq \frac{8\log(2/\delta)}{\epsilon^2}$.

Proof. An application of Theorem 9 with the explanations above and $\lambda\delta = \epsilon$.

6.3 Tradeoffs and practical recommendations

The proposed generalization bounds give worst case estimations of the generalization risk through different approaches. They are helpful in finding connections between the involved complexities of the network and the size of data. This raises several acute tradeoffs that are worth explaining. We derive different tradeoffs that occur under alternative assumptions on the involved parameters.

Many of the tradeoffs include big O notations. The following bound is often applied in order to derive sufficient conditions for the relevant quantities:

$$\forall a \geq 1, b > 0 :\ x \geq 4a\log(2a) + 2b \Rightarrow x \geq a\log(x) + b \tag{5}$$

Tradeoff between k and m Refering to Theorem 11. Assume that the total number of source samples $mk = M$ is fixed. An interesting question is how many samples to invest in each task (i.e, what is the best m).

With no loss of generality, we assume that $8 \leq \sqrt{4|J|\log(2me)}$ and $4 \leq \sqrt{|I|\log(mke)}$. We are interested in bounding the regularization terms with ϵ,

$$\sqrt{\frac{2|I|\log(mke)}{k}} \leq \epsilon, \quad \sqrt{\frac{8|J|\log(2me)}{m}} \leq \epsilon$$

By Equation 5, we derive a sufficient condition,

$$m = \Theta\left(\frac{|J|\log(|J|/\epsilon)}{\epsilon^2}\right) \text{ and } k = \Theta\left(\frac{|I|\log(|I|\cdot|J|/\epsilon)}{\epsilon^2}\right)$$

Therefore,

$$m \approx \Theta\left(\sqrt{M\frac{|J|\log(|J|)}{|I|\log(|I|\cdot|J|)}}\right) \text{ and } k \approx \Theta\left(\sqrt{M\frac{|I|\log(|I|\cdot|J|)}{|J|\log(|J|)}}\right)$$

(Neglecting constants and $\log(1/\epsilon)$).

The need to increase k as a function of m Refering to Theorem 11. It is very natural to believe that whenever m increases, k should also increase. That is because, a selection of B that depends only on very accurate information of k fixed number of tasks is biased. The selected B would fit very well with those tasks but might fail to fit with unseen different tasks. This is an overfitting that might occur only in the case of transfer learning. We would like to measure how much is sufficient to increase k as a function of m in order to avoid overfitting. According to Theorem 10, if we fix all of the parameters except m, it is required to take $k = \Omega(\log(m))$ in order to avoid the discussed overfitting. Therefore, the transferring rule $\arg\min_B \epsilon_S(B, r)$ overfits w.r.t tasks if k is smaller by orders of magnitude than $\log(m)$. This is a desireable situation since in most practical situations k is not tiny w.r.t m.

In the other direction, it does not seem there is dependence between k, m that requires m to increase whenever k does.

Tradeoff between m, k and the capacity of the specific part $|J|$ Refering to Theorem 11. The capacity of the specific architecture $\mathcal{H}_u$ is measured by $|J|$. The VC dimension of $\mathcal{H}_u$ depends only on that capacity, i.e, $\text{vc}(\mathcal{H}_u) = O(|J|\log|J|)$. From the bound, we adress that $m = \Theta(|J|\log|J|)$ is sufficient in order to overcome the size of the specific architecture. In addition, it seems that the dependence of k on $|J|$ is much weaker. In the previous tradeoff, the dependence of k on m is logarithmic, i.e, it is required to take $k = \Omega(\log(m))$ in order to avoid overfitting. Therefore, in the case where m and k are chosen wisely (satisfying $m = \Theta(|J|\log|J|)$ and $k = \Omega(\log(m))$) then $k = \Omega(\log\log|J|)$. A Larger m is required in order to train the specific part of each network separately.

Tradeoff between k and the capacity of the transfer $|I|$ Refering to Theorem 11. As before, we arrive at $k = \Theta(|I|\log|I|)$ is sufficient in order to overcome the size of the transfer architecture. The combination of this argument and the very weak dependence of k on $|J|$ raises the insight that larger k are required mostly to overcome the capacity of the transfer (i.e, $|I|$). Larger k is required to overcome the common transfer architecture. It can also be said that k depends on the whole size of the architecture, but it has a much stronger dependence on the capacity of the transfer despite the specific part.

Tradeoff between k and the number of target samples n By the fundamental Theorem of learnability Vapnik & Chervonenkis (1971), the sample complexity (of a binary classification learning setting) is $\Theta\left(\frac{\text{vc}+\log(1/\delta)}{\epsilon^2}\right)$. Therefore, in order to reduce n, it is necessary to decrease the VC dimension of the post transfer learning setting. By Lemma 7, we have target VC dimension $O(|J|\log|J|)$. Thus, it is desired to increase $|I|$. Nevetheless, by the conclusion of the last tradeoff, it will require k to grow linearly with $|I|\log|I|$.

Transferring too much information hurts performance The richer the source data is, the more information that can be transferred. Transferring too much information hurts performance. This can be seen in the bottleneck effect demonstrated in Taigman et al. (2015), where creating a lower-dim representation improves transfer performance. Bottleneck in the context of information theory was investigated by Tishby et al. (2000), Tishby & Zaslavsky (2015).

We consider the case where J consists of the bottleneck weights (i.e, all weights between the representation layer and the output). In this case, post transfer, the size of target data required to obtain an error rate at most ϵ far from the optimum is $n = \Theta\left(\frac{|J|+\log(1/\delta)}{\epsilon^2}\right)$. Therefore, in order to control the post transfer error rate, we have to require $|J| = \Theta(n\epsilon^2)$ (neglecting $\log(1/\delta)$). The best possible representation, that has the smallest transfer generalization risk, is the one of size equals to the examples size. We are looking for a smaller representation that is still ϵ-close to the best possible representation. More formally, for any ϵ, there is an optimal size for the representation that has error at most ϵ larger than the error of the best representation. The error of the bias learned in the transfer stage depends on how far J of size $\Theta(n\epsilon^2)$ is from optimal representation size.

Learning with noisy labels Refering to Theorem 11. The performance of Convnets to learn tasks with noisy labels was studied by Sukhbaatar & Fergus (2014). They showed that Convnets have good performance in learning tasks even when the labels are noisy. They introduced an extra noise layer that adapts the network to the noise distribution. The proposed factory framework can model noisy labels and shed light on this situation.

When learning with noisy labels, there is a target task $d = (c^*, p)$. The goal is to learn c^* through k random noisy streams with "mean" d. We introduce a transfer learning setting $\mathscr{T} = (T, \mathscr{C}, \mathcal{E})$. The underlying learning task $T = (\mathcal{H}, Z, \ell)$ is a supervised binary classification setting. The hypothesis class is a neural networks architecture $\mathcal{H} := \mathcal{H}_{V,E,\text{sign}}$. The environment is,

$$\mathcal{E} = \{(c, p) \mid c \text{ is any function}\}$$

The factory $\mathcal{D}$ is symmetric around (c^*, p) (i.e, the probability to sample (c_1, p) is equal to the probability to sample (c_2, p) under the assumption that $\epsilon_d(c_1) = \epsilon_d(c_2)$).

In this setting, the learned common representation is the full neural network. It can also be said that the algorithm learns a neural network that fits best with random noisy streams. The hypothesis class family is $\mathscr{C} := \mathcal{H}^I_{V,E,\sigma}$ such that $I = E$, i.e,

$$\mathscr{C} := \mathcal{H}^E_{V,E,\sigma} = \left\{\{h_{V,E,\sigma,w}\} \mid w \in \mathbb{R}^{|E|}\right\}$$

This can be treated simply as $\mathcal{H}_{V,E,\sigma}$. Therefore, each h_B (corresponding to bias $B \in \mathscr{C}$) is simply a neural network in $\mathcal{H}$ (i.e, a concept). In this case, the transfer risk is,

$$\epsilon_{\mathcal{D}}(c) = \mathbb{E}_{b\sim\mathcal{D}}[\epsilon_b(c)]$$

This quantity is minimized by any c such that the set $\{x \mid c(x) \neq c^*(x)\}$ has probability 0 (w.r.t p). Any other function will not minimize this quantity. We apply Theorem 10 (see Appendix F) with $\mathcal{H}_t := \mathcal{H}$ (and $\mathcal{H}_u = \emptyset$). In addition, $|J| = 0$ and $|I| = |E|$. Let $c_S = \arg\min_c \epsilon_S(c) = \frac{1}{k}\sum_{i=1}^k \epsilon_{s_i}(c)$, then with probability $\geq 1 - \delta$ (over S),

$$\epsilon_{\mathcal{D}}(c_S) \leq \inf_{c\in\mathcal{H}} \epsilon_{\mathcal{D}}(c) + \frac{8 + \sqrt{4|E| \cdot \log(2emk)}}{\delta\sqrt{2k}} + \frac{16}{\delta\sqrt{2m}} \tag{6}$$

Therefore, the output c_S converges to the best possible hypothesis as k, m tend to increase.

Accurately, it is sufficient to provide,

$$m = \Theta\left(\frac{1}{(\delta\epsilon)^2}\right) \text{ and } k = \Theta\left(\frac{|E|}{(\delta\epsilon)^2}\log\left(\frac{|E|}{\delta\epsilon}\right)\right)$$

In order to have, $\epsilon_{\mathcal{D}}(c_S) \leq \inf_{c\in\mathcal{H}} \epsilon_{\mathcal{D}}(c) + \epsilon$.

Therefore, the total number of samples sufficient to provide is $\Theta\left(\frac{|E|}{(\delta\epsilon)^4}\log\left(\frac{|E|}{\delta\epsilon}\right)\right)$.

Comparing the binary classification bounds in Baxter (2000) with our VC bound Refering to Theorem 11. In the work of Baxter (2000), they construct multitask generalization bounds (bounds on the source generalization risk for k specified tasks) for deep learning in the binary classification case. In their analysis, they fix k and conclude that the number of samples per task should be:

$$m = O\left(\frac{|H|\log(1/\epsilon)}{\epsilon^2}\right)$$

Where $H \subset V$ is the set of hidden neurons.

On the other hand, with our analysis we arrived to $m = O\left(\frac{|J|\log(|J|/\epsilon)}{\epsilon^2}\right)$ (neglecting $1/\delta$ in both calculations).

In most interesting cases, $|J|\log|J| \ll |H|$ (see Taigman et al. (2014), Taigman et al. (2015), Donahue et al. (2013), Razavian et al. (2014)).

6.3.1 PAC-Bayes tradeoffs

The tradeoffs above were all derived based on the VC bound. Tradeoffs can also be derived from the PAC-Bayes bounds. The PAC-Bayes settings are more general, since arbitrary weights can be transfered, and not just parts of the architectures. This setting is also not limited to binary classification.

Tradeoff between m and the size of the architecture $|E|$ Refering to Theorem 12. The $\mathrm{KL}(\cdot\|\cdot)$ measures the difference between two distributions. There is a direct connection between the KL-divergence and the dimension of the space of the distributions. In the case we investigate, the prior and posterior distributions are Gaussian distributions. For instance, we can assume that the parameters u_{opt} of the optimal posterior over biases, Q^{opt} and v_{opt} of the optimal posterior distributions over concepts (for each task), q^{opt}, were selected i.i.d from some distribution D and obtain,

$$\mathbb{E}_{u_{opt},v_{opt}\sim D^{|E|}}[\mathrm{KL}(Q_q^{opt}\|p)] = A\cdot|E|, \quad \text{s.t} \;\; A := \mathbb{E}_{x\sim D}\left[x^2/2\right]$$

Requiring that the expected specific regularization term for the optimal posterior be at most ϵ with the fact that $\mathbb{E}[\sqrt{X}] \leq \sqrt{\mathbb{E}[X]}$ and the selection $\lambda\delta = \epsilon$,

$$\mathbb{E}_{u_{opt}\sim D^{|I|}}\left[\sqrt{\frac{\mathrm{KL}(q_{opt}\|p) + \log(2m/\lambda\delta)}{2(m-1)}}\right] \leq \sqrt{\frac{A\cdot|E| + \log(2m/\lambda\delta)}{2(m-1)}} \leq \epsilon$$

That simply concludes to,

$$m = \Theta\left(\frac{A\cdot|E| + \log\left(\frac{1}{\epsilon}\right)}{\epsilon^2}\right) \tag{7}$$

Therefore, it is required to increase m linearly as $|E|$ grows. On the other hand, from the bound it does not seem that there is such a strong dependence between k and $|E|$.

Tradeoff between k and the capacity of the transfer $|I|$ Refering to Theorem 12 and Theorem 13. We use the same analysis as before. It is assumed that the parameters u_{opt} of the optimal posterior distribution, Q_{opt}, were selected i.i.d from some distribution D. Thus,

$$\mathbb{E}_{u_{opt}\sim D^{|I|}}[\mathrm{KL}(Q_{opt}\|P)] = A\cdot|I|, \quad \text{s.t} \;\; A := \mathbb{E}_{x\sim D}\left[x^2/2\right]$$

We refer $\sqrt{\frac{\mathrm{KL}(Q\|P)+\log(2k/\delta)}{2(k-1)}}$ as the transfer regularization term and would like to restrict it to be at most ϵ for the optimal posterior. Using the fact that $\mathbb{E}[\sqrt{X}] \leq \sqrt{\mathbb{E}[X]}$,

$$\mathbb{E}_{u_{opt}\sim D^{|I|}}\left[\sqrt{\frac{\mathrm{KL}(Q_{opt}\|P) + \log(2k/\delta)}{2(k-1)}}\right] \leq \sqrt{\frac{A\cdot|I| + \log(2k/\delta)}{2(k-1)}} \leq \epsilon$$

That simply concludes to,

$$k = \Theta\left(\frac{A\cdot|I| + \log\left(\frac{1}{\delta\epsilon}\right)}{\epsilon^2}\right) \tag{8}$$

Therefore, it is required to increase k linearly as $|I|$ grows.

Tradeoff between k and m Assume that the total number of source samples $mk = M$ is fixed. We provide an analysis, based on Theorem 13.

It is assumed that the parameters u_{opt} and v_{opt} of the optimal posterior distributions, Q_{opt} and q_{opt}, were selected i.i.d from some distribution D, we have:

$$\mathbb{E}_{u_{opt}\sim D^{|I|}}[\mathrm{KL}(Q_{opt}\|P)] = \mathbb{E}_{x\sim D}\left[x^2/2\right]\cdot|I| := A\cdot|I|$$

In addition,

$$\mathbb{E}_{u_{opt},v_{opt}\sim D^{|E|}}[\mathrm{KL}(q_{opt}\|p)] = \mathbb{E}_{x\sim D}\left[x^2/2\right]\cdot|E| = A\cdot|E|$$

In order to ensure that the expected regularization term for the optimal posterior will be at most ϵ by the fact that $\mathbb{E}[\sqrt{X}] \leq \sqrt{\mathbb{E}[X]}$ we may require

$$\lambda = \frac{\epsilon}{3\delta},\quad \sqrt{\frac{A\cdot|E| + \log(2m/\lambda\delta)}{2(m-1)}} \leq \frac{\epsilon}{3} \text{ and } \sqrt{\frac{A\cdot|I| + \log(2k/\delta)}{2(k-1)}} \leq \frac{\epsilon}{3}$$

Applying Equation 5 and neglecting $\log(1/\delta\epsilon)$,

$$m = \Theta\left(\frac{A\cdot|E|}{\epsilon^2}\right) \approx \Theta\left(\sqrt{\frac{|E|\cdot M}{|I|}}\right) \text{ and } k = \Theta\left(\frac{A\cdot|I|}{\epsilon^2}\right) \approx \Theta\left(\sqrt{\frac{|I|\cdot M}{|E|}}\right)$$

Comparing the regression bounds in Baxter (2000) with our PAC-Bayes bound Refering to Theorem 13. In the work of Baxter (2000), they construct transfer generalization bounds for deep learning in regression settings. The bottom line of their analysis concludes that (neglecting $\log(1/\epsilon\delta)$),

$$k = O\left(|E|/\epsilon^2\right) \text{ and } m = O\left(|H|/\epsilon^2\right)$$

Where $H \subset V$ is the set of hidden neurons.

On the other hand, neglecting $\log(1/\epsilon\delta)$, with the analysis above we arrived at $k = O\left(|I|/\epsilon^2\right)$ and $m = O\left(|E|/\epsilon^2\right)$. Therefore, our bound requires fewer number of multiple tasks but more samples per task.

7 Related Work

The standard assumption in supervised machine learning algorithms is to have models trained and tested on samples drawn from the same probability distribution. Often, however, there are many labeled training samples from a source task and the goal is to construct a learning rule that performs well on a target task with a different distribution and little labeled training data. This is the problem of Domain Adaptation (DA), where a successful scheme typically utilizes large unlabeled samples from both tasks to adapt a source hypothesis to the target task. Kifer et al. (2004); Ben-David et al. (2007); Mansour et al. (2009a); Ben-David et al. (2010a) suggest adapting these tasks by considering the divergence between the source and target sources. Based on this divergence, they provide PAC-like generalization bounds. Alternatively, Li & Bilmes (2007) measure the adaption using divergence priors and learn a hypothesis for the target task by applying an ERM rule on the source task. A different approach is due to Yang & Hospedales (2014). They coined a new term called semantic descriptors. These are generic descriptors uniform for a few tasks that reduce the uncertainty about the tasks. Hardness results for DA are explored in Ben-David et al. (2010b).

Our work does not assume that the tasks are comparable, e.g., by divergence of tasks. Our only restriction is having enough data from a common source (the factory), similar to the original PAC model. Two types of common sources are explored. The first is used to investigate the adversary situations when the concepts are selected almost arbitrarily. The second, uses random concepts, and was previouslty proposed by Baxter (2000) in the concept of inductive bias learning, in which there is no one dedicated target task. Our random concept factory differs from that of Baxter (2000) in that the transfer task might have considerably less training examples than the source tasks. We are, therefore, able to model the case in which the source tasks have practically unrestricted samples, while harvesting samples for the target task is much harder. We also discuss common aspects as

in Ando et al. (2005). Their work proposes the Joint ERM rule (which we redefine as the C-ERM rule). In our work, we extend the discussion on this transferring rule and suggest a regularized random version of it derived from PAC-Bayesian bounds introduced in the paper.

Cortes et al. (2008); Crammer et al. (2008); Mansour et al. (2009b) combine several training sources to better describe the target distribution. It is assumed that the distribution of the target task is a linear combination (that sums to 1) of the source distributions. Our work differs from these works, since we do not seek to approximate the target distribution from the multiple sources, but rather to transfer a concept that facilitates learning from a few examples in the target task.

Transfer learning has attracted considerable recent attention, with the emergence of transfer learning in visual tasks Krizhevsky et al. (2012); Girshick et al. (2014); Fei-Fei et al. (2006); Yang et al. (2007); Orabona et al. (2009); Tommasi et al. (2010); Kuzborskij et al. (2013). In these contributions, the application of transfer learning is done without assuming any knowledge about the relatedness of the source and target distributions. Although this setting has been explored empirically with success, a formal theory of transfer learning is mostly missing.

Recently, the generalization properties of the transfer learning approach was investigated in Pentina & Lampert (2014); Kuzborskij & Orabona (2013); Tommasi et al. (2014). Specifically, Pentina & Lampert (2014) measure the representation transfer by PAC-Bayesian generalization. Their approach assumes that the data set of the target task is proportional to each source data set. Alternatively, Kuzborskij & Orabona (2013); Tommasi et al. (2014) measure the amount of transfer according to its stability with respect to the leave-one-out error Mukherjee et al. (2002),Bousquet & Elisseeff (2002). Our work differs from these works by its scope. We focus on presenting when transfer learning is meaningful and how to measure its success when transferring a narrowed hypothesis class between source and target tasks. This process requires a target data set that is smaller than each source data set. Also, Hardt et al. (2015) showed that under appropriate assumptions, the SGD learning algorithm Rumelhart et al. (1988) is uniformly stable (when replacing one sample with another). In their setting, the aim is solving only one task at a time. They assume convexity of the loss function (w.r.t to the hypothesis). Their intention is to apply the results to the case of neural networks.

One of the main themes of the work is describing approaches for learning common representations between multiple similar tasks. We apply our general mechanisms based on VC and PAC-Bayes theories for the special case of deep learning. In our model, we select a set of representations and output one that seems to fit the data. A different approach for learning representations is based on invariants and selectivity of representations. This perspective appears in the work of Poggio et al. (2015). It is shown that representations that are both invariant to transformations and at the same time selective, can decrease the required amount of data.

One of the conjectures concerning deep learning is that deep neural networks have great generalization ability. One way to tackle this question is by claiming that neural networks have excellent transferability. In the work of Yosinski et al. (2014), they show empirically that neurons of the first few layers appear to be more general than the last layers, which are more task oriented. In our work, we consider this same question. We decompose the network into a transferred part and a specific part, i.e., the first layers and last layers. In our VC-style transfer generalization bound, the regularization decomposes between transfer and specific regularizations. The transfer regularization penalizes by the size of the transfer part over the number of source tasks; and the specific regularization penalizes by the size of the specific part.

The work of Yosinski et al. (2014) also studies the notion of co-adaptation, i.e., fine-tuning of the transferred layers. Such fine-tuning is not part of our framework, since it is hard to characterize the amount of transfer left after this process takes place. In practice, one balances between adapting the transferred part and the specific part by employing multiple learning rates. A stability framework might be more suitable for studying such effects, since it models the training process.

In this work, we also refer to the problem of learning with noisy labels. This problem was discussed by Sukhbaatar & Fergus (2014), where it was shown that Convnets have good performance in learning tasks even when the labels are noisy. Our theoretical model is able to derive a suitable generalization bound for such training by creating noisy factories.

8 Conclusions

We generalize the notion of PAC-learning. Whereas in PAC-learning, it is assumed that train and test samples are drawn from the same distribution, in our framework we consider a hierarchical model, in which the sample distributions $\{d_i\}_{i=1}^k \cup \{d_t\}$ are drawn from a top-level distribution. In addition, most of our results are concerned with the case in which all k source concepts $\{c_i\}_{i=1}^k$, as well as the target concept c_t are selected from the same unknown subclass. At first, we discussed the case where the concepts are selected arbitrarily and then we turned to the case where the concepts are selected i.i.d along to the distributions.

Our results have direct implications to the practice of transfer learning using neural networks. We model multi-layered networks and the transfer process that is often practiced with such networks. Using the generalization bounds that we obtain for transfer learning, we are able to derive various trade-offs that link quantities such as network size, size of learned representations, and the required amount of source and target data.

This factory based construction can be applied recursively, and results in distributions of distributions of distributions and in hypothesis classes that are divided hierarchically. Most generally, a tree structure might link various source tasks, with different levels of relatedness. Such a model, in addition to its recursive simplicity, might be a good model for ecological learning in which one learner performs continuous learning; each additional task encountered promotes the learning of future tasks.

Acknowledgments

We would like to thank Tomaso Poggio, Yishay Mansour and Ronitt Rubinfeld for illuminating discussions during the preparation of this paper. This research was partly supported by a Grant from the GIF, the German-Israeli Foundation for Scientific Research and Development.

References

Ando, Rie Kubota, Zhang, Tong, and Bartlett, Peter. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

Baxter, Jonathan. A model of inductive bias learning. *J. Artif. Intell. Res. (JAIR)*, 12:149–198, 2000.

Ben-David, Shai, Blitzer, John, Crammer, Koby, Pereira, Fernando, et al. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137, 2007.

Ben-David, Shai, Blitzer, John, Crammer, Koby, Kulesza, Alex, Pereira, Fernando, and Vaughan, Jennifer Wortman. A theory of learning from different domains. *Machine learning*, 79(1-2): 151–175, 2010a.

Ben-David, Shai, Lu, Tyler, Luu, Teresa, and Pál, Dávid. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pp. 129–136, 2010b.

Blumer, Anselm, Ehrenfeucht, A., Haussler, David, and Warmuth, Manfred K. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

Bousquet, Olivier and Elisseeff, André. Stability and generalization. *J. Mach. Learn. Res.*, 2:499–526, March 2002. ISSN 1532-4435.

Cortes, Corinna, Mohri, Mehryar, Riley, Michael, and Rostamizadeh, Afshin. Sample selection bias correction theory. In *Algorithmic learning theory*, pp. 38–53. Springer, 2008.

Crammer, Koby, Kearns, Michael, and Wortman, Jennifer. Learning from multiple sources. *The Journal of Machine Learning Research*, 9:1757–1774, 2008.

Donahue, Jeff, Jia, Yangqing, Vinyals, Oriol, Hoffman, Judy, Zhang, Ning, Tzeng, Eric, and Darrell, Trevor. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

Fei-Fei, Li, Fergus, Robert, and Perona, Pietro. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611, 2006.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, and Malik, Jitendra. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 580–587. IEEE, 2014.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. *ArXiv e-prints*, September 2015.

Kakade, Sham M. and Tewari, Ambuj. VC dimension of multilayer neural networks, range queries. Lecture notes, 2008.

Kifer, Daniel, Ben-David, Shai, and Gehrke, Johannes. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 180–191. VLDB Endowment, 2004.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Kuzborskij, Ilja and Orabona, Francesco. Stability and hypothesis transfer learning. In *Proceedings of The 30th International Conference on Machine Learning*, pp. 942–950, 2013.

Kuzborskij, Ilja, Orabona, Francesco, and Caputo, Barbara. From n to n+ 1: Multiclass transfer incremental learning. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3358–3365. IEEE, 2013.

Li, Xiao and Bilmes, Jeff. A Bayesian divergence prior for classiffier adaptation. In *International Conference on Artificial Intelligence and Statistics*, pp. 275–282, 2007.

Mansour, Yishay, Mohri, Mehryar, and Rostamizadeh, Afshin. Domain adaptation: Learning bounds and algorithms. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009a.

Mansour, Yishay, Mohri, Mehryar, and Rostamizadeh, Afshin. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pp. 1041–1048, 2009b.

McAllester, D. A. Some PAC-Bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (Madison, WI, 1998)*, pp. 230–234. ACM, 1998.

McAllester, David. A pac-bayesian tutorial with A dropout bound. *CoRR*, abs/1307.2118, 2013.

Mukherjee, Sayan, Niyogi, Partha, Poggio, Tomaso, and Rifkin, Ryan. Statistical learning: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. Technical report, Advances in Computational Mathematics, 2002.

Orabona, Francesco, Castellini, Claudio, Caputo, Barbara, Fiorilla, Angelo Emanuele, and Sandini, Giulio. Model adaptation with least-squares SVM for adaptive hand prosthetics. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 2897–2903. IEEE, 2009.

Pentina, Anastasia and Lampert, Christoph H. A pac-bayesian bound for lifelong learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 991–999, 2014.

Poggio, Tomaso, Anselmi, Fabio, and Rosasco, Lorenzo. On invariance and selectivity in representation learning. *Information and Inference: A Journal of the IMA*, 2015.

Razavian, Ali Sharif, Azizpour, Hossein, Sullivan, Josephine, and Carlsson, Stefan. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pp. 696–699. MIT Press, 1988.

Sauer, N. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145 – 147, 1972.

Shalev-Shwartz, Shai and Ben-David, Shai. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

Shalev-shwartz, Shai, Shamir, Ohad, Srebro, Nathan, and Sridharan, Karthik. Learnability, stability and uniform convergence. *JMLR*, 2010.

Sukhbaatar, Sainbayar and Fergus, Rob. Learning from noisy labels with deep neural networks. *CoRR*, abs/1406.2080, 2014.

Taigman, Yaniv, Yang, Ming, Ranzato, Marc'Aurelio, and Wolf, Lior. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Taigman, Yaniv, Yang, Ming, Ranzato, Marc'Aurelio, and Wolf, Lior. Web-scale training for face identification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 2746–2754, 2015.

Tishby, Naftali and Zaslavsky, Noga. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pp. 1–5, 2015.

Tishby, Naftali, Pereira, Fernando C. N., and Bialek, William. The information bottleneck method. *CoRR*, physics/0004057, 2000.

Tommasi, Tatiana, Orabona, Francesco, and Caputo, Barbara. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3081–3088. IEEE, 2010.

Tommasi, Tatiana, Orabona, Francesco, and Caputo, Barbara. Learning categories from few examples with multi model knowledge transfer. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(5):928–941, 2014.

Valiant, L. G. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.

Vapnik, V. N. and Chervonenkis, A. Ya. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Vapnik, Vladimir. *Statistical learning theory*. Wiley, 1998.

Vapnik, Vladimir N. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

Yang, Jun, Yan, Rong, and Hauptmann, Alexander G. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th international conference on Multimedia*, pp. 188–197. ACM, 2007.

Yang, Yongxin and Hospedales, Timothy M. A unified perspective on multi-domain and multi-task learning. *CoRR*, abs/1412.7489, 2014.

Yosinski, Jason, Clune, Jeff, Bengio, Yoshua, and Lipson, Hod. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

A Proofs for the claims in Section 4

Theorem 3. The proof will be separated into two parts. The first, proving the easier directions; (3) $\implies$ (1), (1) $\implies$ (2). The second, proving the less trivial directions (2) $\implies$ (1) and (1) $\implies$ (3).

(3) $\implies$ (1): if the class is PAC-learnable, then it has a finite VC dimension. In particular, it has a finite transfer VC dimension (with A that always return $\mathcal{H}$).

(1) $\implies$ (2): denote the transfer VC dimension v. We claim that a T-ERM rule transfer learns the hypothesis class.

We require k, m enough to achieve both $\inf_{c\in B} \epsilon_{d_t}(c) \leq \epsilon/2$ and $\text{vc}(B) \leq v$ by the narrowing with probability $\geq 1 - \delta/2$. By Lemma 1, if we require $n \geq N_v(\epsilon/2, \delta/2)$, an ERM rule would output c_{out} that has error $\epsilon_{d_t}(c_{out}) \leq \inf_{c\in B} \epsilon_{d_t}(c) + \epsilon/2$ with probability $\geq 1 - \delta/2$ for any hypothesis class of VC dimension at most v generated by the narrowing. Therefore, by union bound, both events hold with probability $\geq 1 - \delta$ and we have,

$$\mathbb{P}_{S\sim\mathcal{D}[k,m,n]}[\epsilon_{d_t}(c_{out}) \leq \epsilon/2 + \epsilon/2 = \epsilon] \geq 1 - \delta$$

This procedure transfer learns the hypothesis class for the required k, m as noted and any $n \geq N_v(\epsilon/2, \delta/2)$.

(2) $\implies$ (1): before we prove this direction, we adopt several definitions. The goal of this process is to define a worst case version of transfer VC-style dimensions (it will be named, w-dimension). We will prove that if a hypothesis class is PAC-transferable, then it has a finite w-candidate. In addition, a further statement shows that having a finite w-dimension yields having a finite VC dimension.

Definition 8. *1. (Conditionally shattered set). We say that $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$ conditionally shatters the data set $(s_{[1,k]}, o_t)$ (or alternatively, shatters s_t given $o_{[1,k]}$ - a labeled source data) if for all labelings of o_t there exists $K \in \mathscr{C}$ such that there are concepts $c_t, c_1, ..., c_k \in K$ that satisfy the following conditions:*

- *c_i differs from c_t for all $i \in [k]$.*
- *c_i is consistent with s_i for all $i \in [k]$.*
- *c_t is consistent with the labeling selected for o_t.*

2. (Transfer shattering). We say that $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$ transfer shatters the unlabeled data $O = (o_{[1,k]}, o_t)$ if there exists a labeling for $o_{[1,k]}$, denoted $s_{[1,k]}$, such that $(s_{[1,k]}, o_t)$ is conditionally shattered by $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$.

The next step is to define a worse-case transfer VC dimension of a hypothesis class. We will call these w-candidates. The dimension will be an un-tight bound on the potential dimension.

Definition 9 (w-dimension)**.** *A hypothesis class $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$ has finite w-dimension if there is a tuple (d_0, d_1, d_2) such that any $O = (o_{[1,k]}, o_t)$ with sizes $|o_i| = d_1$ and $|o_t| = d_2$ and $k = d_0$, is not transfer shattered by $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$. Otherwise, we say that it has infinite w-dimension. The tuple (d_0, d_1, d_2) is called w-candidate.*

Lemma 8. *If $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$ is a hypothesis class of binary classifiers which is PAC-transferable, then it has finite w-dimension (at least one w-candidate).*

Proof. Assume by contradiction that the w-dimension of $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$ is infinite, and that $\mathcal{H}$ is transferable. Let A be the learning algorithm, that requires k source tasks and m, n source and target examples to obtain accuracy $\epsilon = 0.1$ and confidence $1 - \delta = 0.9$. In other words, after seeing m examples from each one of k source tasks and n examples from the target, A outputs a hypothesis $A(S) \in \mathcal{H}$ that satisfies:

$$\mathbb{P}_S[\epsilon_{d_t}(A(S)) \leq 0.1] > 0.9$$

Since $\mathcal{H}$ has no w-candidates, there exists a data set $O = (o_{[1,k]}, o_t)$ that is transfer shattered by $\mathcal{H}$ such that $o_i = o_i' \cup o_t$ is a disjoint union of o_i' of size m and o_t (for all $i \in [k]$). In addition, o_t is of

size $2(n+mk)$. We define distributions over $\mathcal{X}$ as follows:

$$p_i = \begin{cases} x \in o'_i & \text{with probability of } \frac{1}{m} \\ x \notin o'_i & \text{with probability of } 0 \end{cases}, \quad p_t = \begin{cases} x \in o_t & \text{with probability of } \frac{1}{2(n+mk)} \\ x \notin o_t & \text{with probability of } 0 \end{cases}$$

Because O is transfer shattered, there is a labeled version $s_{[1,k]}$ of $o_{[1,k]}$ such that for all labelings of o_t there is $c \in \mathcal{H}$ that is consistent with o_t and shares $K \in \mathscr{C}$ with $c_1, ..., c_k$ that are consistent with $s_{[1,k]}$ (respectively). The factory (distribution over distributions):

$$\mathcal{D} = \begin{cases} p_t & \text{with probability of } 1 - \delta/2 \\ p_i;\ i \in [k] & \text{with probability of } \frac{\delta}{2k} \end{cases}$$

The factory selects c_t at first,

$$c_t(x) = \begin{cases} 1; x \in o_t & \text{with probability of } 1/2 \\ 0; x \in o_t & \text{with probability of } 1/2 \\ 0; x \notin o_t & \text{with probability of } 1 \end{cases}$$

The factory then selects i.i.d $k+1$ distributions from $\mathcal{D}$ and the concepts $c_1, ..., c_k$ that differ from c_t and are consistent with $s_{[1,k]}$ (respectively). A selects i.i.d m examples from each of the selected source tasks and obtains a source data that is $\subset s_{[1,k]}$ (for the i'th task, it has data set $\subset s_j$ for some $j \in [k]$) and n examples from the target task. It outputs a concept $A(S)$ to approximate the target concept (it is consistent in the best case). We notice that by definition of $\mathcal{D}$, the target task is $d_t = (c_t, p_t)$ with probability $= 1 - \delta/2$. The algorithm receives at most half of the examples in o_t. By conditional shattering of $(s_{[1,k]}, o_t)$, for the given labeling of the source part of the data, any target concept on o_t is possible. The target concept was selected randomly with coin flips and if $d_t = (c_t, p_t)$ is the target task, the probability for mistake is $\geq 1/4$ (since the probability for mistake in the half unseen examples is $1/2$). Therefore, we result with an output hypothesis that has a probability of mistake $\geq 1/4$ with probability $\geq 1 - \delta/2 = 0.95$. The expected error of $A(S)$ is:

$$\mathbb{E}_S[\epsilon_{d_t}(A(S)) \mid d_t = (c_t, p_t)] \geq n \cdot 0 \cdot \frac{1}{2(mk+n)} + (mk+n) \cdot \frac{1}{2} \cdot \frac{1}{2(mk+n)} = 1/4$$

By the conditional expectation rule, we get $\mathbb{E}_S[\epsilon_{d_t}(A(S))] \geq 0.95 \cdot 0.25 = 0.237$. On the other hand, by transferability we observe that:

$$\mathbb{E}_S[\epsilon_{d_t}(A(S))] \leq 0.9 \cdot 0.1 + 0.1 \cdot 1 = 0.19 < 0.237$$

This contradicts the above evaluation. □

(1) $\implies$ (3): we have concluded the proof that PAC-transferability yields the existence of a finite w-dimension. The following Lemma claims that a hypothesis class that has a finite w-dimension also has a finite VC dimension (which is equivalent to PAC-learnability by the fundamental Theorem of learning). This will conclude that a PAC-transferable hypothesis class is also PAC-learnable in the binary classification case.

Theorem 14. *Let $\mathcal{H} = \bigcup_{B \in \mathscr{C}} B$ be a hypothesis class with w-candidate (a_0, a_1, a_2). Then $\mathcal{H}$ has VC dimension $\leq 4a_2 \log(2a_2) + 2(a_0+1)(a_1+1) \cdot \log(2)$. In particular, if $\mathcal{H}$ has finite w-dimension, it has finite VC dimension.*

Proof Let $o_{[1,k]}$ be an unlabeled source data. We denote $o^y_{[1,k]}$ the same source data set labeled by the vector $y \in \{0,1\}^{mk}$. In addition, $Q(o^y_{[1,k]})$ will denote the set of all $c \in \mathcal{H}$ such that there exist $c_1, ..., c_k$ consistent with $o^y_{[1,k]}$ (i.e, c_i is consistent with the labeling of o_i under y) and $K \in \mathscr{C}$ for which $c, c_1, ..., c_k \in K$. Assume the class $\mathcal{H} = \bigcup_{B \in \mathscr{C}} B$ has w-candidate (a_0, a_1, a_2). Then, for all source data sets $o_{[1,k]}$ with $k \geq a_0 + 1$ and $m \geq a_1 + 1$ we have that $Q(o^y_{[1,k]})$ has VC dimension at most a_2 for all labelings y of $o_{[1,k]}$. Nevertheless, $\mathcal{H} = \bigcup_y Q(o^y_{[1,k]})$ (the union is over the different labelings of the source data). There are 2^{mk} different such labelings, y. The VC dimension of a union of r hypothesis classes with VC dimension at most a is $\leq 4a \log(2a) + \log(r)$. Thus, when taking $k = a_0 + 1, m = a_1 + 1$, we determine that the VC dimension of $\mathcal{H}$ is bounded by the expression above.

□

Lemma 2. The proof relies on the fact that $\{\text{sign}(\sin(hx))\}_{h\in\mathbb{R}}$ is not learnable with 0-1 loss. We consider the following hypothesis class $\mathcal{H} = \bigcup_{h\in\mathbb{R}} H_h$ where:

$$\mathscr{C} = \{H_h : \forall h \in \mathbb{R} \setminus \{0\}\}, \quad \text{s.t} \quad H_h = \{hx, \text{sign}(\sin(hx))\}$$

$\{hx, \text{sign}(\sin(hx))\}$ is a set of two functions; the linear function $a(x) = hx$ and the function $b(x) = \text{sign}(\sin(hx))$. The hypothesis class $\mathcal{H}$ is not PAC-learnable with respect to the squared loss. If there were a learning algorithm that learns $\mathcal{H}$ with respect to the squared loss, one could simulate a learning algorithm for $\{\text{sign}(\sin(hx))\}$ by just applying the same learner. This is equivalent to the learnability of $\{\text{sign}(\sin(hx))\}$ with respect to 0-1 loss, because squared loss and 0-1 loss are the same in the binary classification case. But, $\{\text{sign}(\sin(hx))\}$ is not learnable with respect to 0-1 loss and we conclude that $\mathcal{H}$ is not PAC-learnable with respect to the squared loss. Nevertheless, $\mathcal{H} = \bigcup_{h\in\mathbb{R}} H_h$ is PAC-transferable. We use the following procedure,
Input: $S \sim \mathcal{D}[k, m, n]$.
Output: a hypothesis c_{out} such that $\epsilon_{d_t}(c_{out}) \leq \epsilon$ with probability $\geq 1 - \delta$.

Case a If there are at least two different samples (x_1, y_1) and (x_2, y_2) in the same source data set s_i:

1. If $y_1 = 0$ and $x_1, x_2 \neq 0$: return a consistent hypothesis hx with s_t.
2. If $x_1 = y_1 = 0$: then recognize $h = y_2/x_2$ and return $\text{sign}(\sin(hx))$.
3. If $y_1 = y_2 = 1$: then return a consistent hypothesis hx with s_t.
4. Else (if $y_1 \neq y_2$ and $y_1, y_2 \neq 0$): recognize h and return $\text{sign}(\sin(hx))$.

Case b Else, return $c \equiv y$, where y is the label of the first target sample.

First we denote $A_{\epsilon,\delta}$, the set of all distributions p that have point $x \in \mathbb{R}$ such that $prob_p(x) \geq \max\{1 - \epsilon, \sqrt{1-\delta}\}$. In addition, n will be the number of samples required to learn the hypothesis class $\{hx : h \in \mathbb{R} \setminus \{0\}\}$ (w.r.t 0-1 loss) with error rate at most ϵ and confidence rate $\sqrt{1-\delta}$.

If case a is satisfied; In this part of the proof, we show that in each case, the output of the algorithm has error at most ϵ with probability at least $1 - \delta$. It is obvious that all of the presented options cover all the cases for a.

Case a.2; since $(x_1, 0) = (0, 0)$ occurs only for hx, then the source concept is hx with $h := y_2/x_2$. Thus, by the definition of the factory, the target concept must be $\text{sign}(\sin(hx))$.

Case a.1 or a.3; the source concept is of the form $\text{sign}(\sin(hx))$. Thus, we employ a selection of hx that is consistent with the target data set. The output, in this case, has error at most ϵ with probability $\geq \sqrt{1-\delta}$.

Case a.4; the source concept is of the form hx (since at least one label is $y_i \neq 0, 1$). In addition, provided with the samples (x_1, y_1) and (x_2, y_2) one is able to recognize h easily. Thus, the concrete target concept is returned.

If $prob_{\mathcal{D}}(A_{\epsilon,\delta}) \geq \sqrt{1-\delta}$**;** then with probability $\geq \sqrt{1-\delta}$ a selected task d has probability $\geq 1 - \epsilon$ to draw a unique sample. If case a is satisfied, it was already shown that the probability for the desired output is $\geq \sqrt{1-\delta} \geq 1 - \delta$. On the other hand, if case b is satisfied, with probability $\geq \sqrt{1-\delta}^2 = 1 - \delta$, the error rate of the output with respect to d_t is $\leq \epsilon$.

Otherwise, in the case where $prob_{\mathcal{D}}(A_{\epsilon,\delta}) \leq \sqrt{1-\delta}$**;** the probability of having at least one task from which at least two different samples are taken is at least $(1 - \sqrt{1-\delta}^k) \cdot (1 - \max\{1-\epsilon, \sqrt{1-\delta}\}^m)$ which is larger than $\sqrt{1-\delta}$ for large enough m and k functions of ϵ, δ. Therefore, case a holds with probability $\geq \sqrt{1-\delta}$ and the algorithm outputs the desired output with probability $\geq \sqrt{1-\delta}^2 = 1 - \delta$.

□

Lemma 3. We consider an example in which $\mathscr{C} = \{E_0, E_1\}$. Let $\mathcal{H} = E_0 \cup E_1$ be the set of all boolean functions on n variables, where E_0 is the set of all functions that map $(0, ..., 0)$ to 0, and

$E_1 = \mathcal{H} \setminus E_0$. As we show next, this hypothesis class has a transfer VC dimension $\leq 2^n - 1$ and VC dimension 2^n. We suggest the following procedure N,

Input: $S = s_{[1,k]} \sim \mathcal{D}[k, m]$.
Output: hypothesis class $N(S)$ such that $\inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \epsilon$ with probability $\geq 1 - \delta$.
Narrowing apply the following procedure.

Case a If $(0, ..., 0) \in s_i$ for some i - return E_j, where j is the label of $(0, ..., 0)$.

Case b Else, return E_0.

First, the VC dimension of the output $\leq 2^n - 1$. It follows immediately from the size of each class: $|E_0| = |E_1| = 2^{2^n - 1}$ and the fact that the VC dimension of a set of size s is at most $\log_2(s)$. We are left to show that with appropriate functions k, m of ϵ, δ, we have output that satisfies $\inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \epsilon$, with probability $\geq 1 - \delta$. We define A_ϵ to be the set of all distributions p such that $prob_p(0, ..., 0) \leq \epsilon$.

If $prob_{\mathcal{D}}(A_\epsilon) \leq 1 - \delta$**;** then with probability $\geq \delta$ a selected task d has probability $\geq \epsilon$ to draw $(0, ..., 0)$. Thus, in this case, with probability $\geq (1 - (1 - \delta)^k)(1 - (1 - \epsilon)^m)$ at least one sample $(0, ..., 0)$ is drawn from at least one source task. This quantity is larger than $1 - \delta$ for $m > \frac{\log(1 - \sqrt{1 - \delta})}{\log(1 - \epsilon)}$ and $k > \frac{\log(1 - \sqrt{1 - \delta})}{\log(\sqrt{1 - \delta})}$. Therefore, the output $A(S)$ satisfies $\inf_{c \in N(S)} \epsilon_{d_t}(c) = 0$ with probability $\geq 1 - \delta$.

Otherwise, in the case where $prob_{\mathcal{D}}(A_\epsilon) \geq 1 - \delta$**;** the probability of d_t such that $prob_{d_t}(0, ..., 0) \leq \epsilon$ is at least $1 - \delta$. Thus, if case b is satisfied, we have $\inf_{c \in N(S)} \epsilon_{d_t}(c) = 0 \leq \epsilon$ if E_0 is the subject of the factory. Otherwise, $\inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \epsilon$ since $prob_{d_t}(0, ..., 0) \leq \epsilon$ with probability $\geq 1 - \delta$. Nevertheless, if case a of the narrowing method is still satisfied, then since $c_t \in N(S)$, we have $\inf_{c \in N(S)} \epsilon_{d_t}(c) = 0 \leq \epsilon$.

We conclude that the program returns $N(S)$ that satisfies $\inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \epsilon$ with probability $\geq 1 - \delta$ (for k, m as above).

□

B Proofs for the claims in Section 4.3

We begin with a version of Theorem 1.

$$\forall p, B, K : \mathbb{E}_{o\sim p^m}\left[\sup_{c_1\in B, c_2\in K}\left|\epsilon_{p_i}(c_1,c_2)-\epsilon_o(c_1,c_2)\right|\right] \le \frac{4+\sqrt{\log(\tau_B(2m))+\log(\tau_K(2m))}}{\sqrt{2m}} \tag{9}$$

Proof. This is an extension of Theorem 1. The proof can be found in (cf. Shalev-Shwartz & Ben-David (2014), Theorem 6.11). We only explain the modifications in the proof.

A union bound is taken over the set of all $c \in \mathcal{H}_{o\cup o'}$ (such that o and o' are two unlabeled data sets of size m). In our case, since the arbitrary selection of c_1, c_2, we take supremum over $c_1 \in B_{o\cup o'}$ and $c_2 \in K_{o\cup o'}$ instead. This is a multiplicative blowup in the number of configurations. In the worst case, there are $\tau_B(2m)\cdot\tau_K(2m)$ configurations. □

Theorem 4. For a set $U = \{p_1, ..., p_k\}$ we denote $E_U(B,K) = \frac{1}{k}\sum_{i=1}^k \mathrm{err}_{p_i}(B,K)$. We note that $\mathbb{E}_{U\sim\mathcal{D}[k]}[E_U(B,K)] = E(B,K)$. Therefore, we have:

$$\mathbb{E}_{O\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|E(B,K)-E_O(B,K)\right|\right] = \mathbb{E}_{O\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\mathbb{E}_{U\sim\mathcal{D}[k]}[E_U(B,K)-E_O(B,K)]\right|\right]$$

By the triangle inequality and the fact that $\sup_c \mathbb{E}[X_c] \le \mathbb{E}[\sup_c X_c]$ we get:

$$\mathbb{E}_{O\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|E(B,K)-E_O(B,K)\right|\right] \le \mathbb{E}_{O\sim\mathcal{D}[k,m],U\sim\mathcal{D}[k]}\left[\sup_{B\in\mathscr{C}}\left|E_U(B,K)-E_O(B,K)\right|\right]$$

Next, we show that $\left|\mathrm{err}_p(B,K)-\mathrm{err}_{o'}(B,K)\right| \le \sup_{c_1,c_2}|\epsilon_p(c_1,c_2)-\epsilon_{o'}(c_1,c_2)|$.

For an unlabeled data set o,

$$\begin{aligned}
&\forall c_1,c_2 : \epsilon_o(c_1,c_2) \le \epsilon_p(c_1,c_2) + |\epsilon_o(c_1,c_2)-\epsilon_p(c_1,c_2)| \\
&\implies \forall c_1,c_2 : \epsilon_o(c_1,c_2) \le \epsilon_p(c_1,c_2) + \sup_{c_1}|\epsilon_o(c_1,c_2)-\epsilon_p(c_1,c_2)| \\
&\implies \forall c_2 : \inf_{c_1}\epsilon_o(c_1,c_2) \le \inf_{c_1}\epsilon_p(c_1,c_2) + \sup_{c_1}|\epsilon_o(c_1,c_2)-\epsilon_p(c_1,c_2)|
\end{aligned}$$

$$\begin{aligned}
\implies \sup_{c_2}\inf_{c_1}\epsilon_o(c_1,c_2) &\le \sup_{c_2}[\inf_{c_1}\epsilon_p(c_1,c_2) + \sup_{c_1}|\epsilon_o(c_1,c_2)-\epsilon_p(c_1,c_2)|] \\
&\le \sup_{c_2}\inf_{c_1}\epsilon_p(c_1,c_2) + \sup_{c_1,c_2}|\epsilon_o(c_1,c_2)-\epsilon_p(c_1,c_2)|
\end{aligned}$$

We obtain: $\sup_{c_2}\inf_{c_1}\epsilon_o(c_1,c_2) - \sup_{c_2}\inf_{c_1}\epsilon_p(c_1,c_2) \le \sup_{c_1,c_2}|\epsilon_o(c_1,c_2)-\epsilon_p(c_1,c_2)|$.

On the other hand, if we denote $c_{1,2} = \arg\min_c \epsilon_o(c,c_2)$ then,

$$\forall o, c_2 : \inf_{c_1}\epsilon_p(c_1,c_2) \le \epsilon_p(c_{1,2},c_2) \le \epsilon_o(c_{1,2},c_2) + |\epsilon_p(c_{1,2},c_2)-\epsilon_o(c_{1,2},c_2)|$$

In particular,

$$\begin{aligned}
&\forall o, c_2 : \inf_{c_1}\epsilon_p(c_1,c_2) \le \inf_{c_1}\epsilon_o(c_1,c_2) + \sup_{c_1}|\epsilon_p(c_1,c_2)-\epsilon_o(c_1,c_2)| \\
&\implies \forall o : \sup_{c_2}\inf_{c_1}\epsilon_p(c_1,c_2) \le \sup_{c_2}\inf_{c_1}\epsilon_o(c_1,c_2) + \sup_{c_1,c_2}|\epsilon_p(c_1,c_2)-\epsilon_o(c_1,c_2)|
\end{aligned}$$

We conclude that, $\left|\mathrm{err}_p(B,K)-\mathrm{err}_{o'}(B,K)\right| \le \sup_{c_1,c_2}|\epsilon_p(c_1,c_2)-\epsilon_{o'}(c_1,c_2)|$.

Let $O' = (o'_1, ..., o'_k)$ be $o'_i \sim p_i^m$ for all $i \in [k]$ where $U = \{p_1, .., p_k\}$, then by the triangle inequality and Equation 9,

$$\begin{aligned}
\forall U, B, K : \mathbb{E}_{O'}\left[\left|E_U(B,K) - E_{O'}(B,K)\right||U\right] &= \mathbb{E}_{O'}\left[\left|\frac{1}{k}\sum_{i=1}^{k}\mathrm{err}_{p_i}(B,K) - \mathrm{err}_{o'_i}(B,K)\right||U\right] \\
&\leq \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{o'_i\sim p_i^m}\left[\left|\mathrm{err}_{p_i}(B,K) - \mathrm{err}_{o'_i}(B,K)\right||U\right] \\
&\leq \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{o'_i\sim p_i^m}\left[\sup_{c_1,c_2}\left|\epsilon_p(c_1,c_2) - \epsilon_{o'}(c_1,c_2)\right||U\right] \\
&\leq \frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{o'_i\sim p_i^m}\left[\sup_{c_1,c_2}\left|\epsilon_p(c_1,c_2) - \epsilon_{o'}(c_1,c_2)\right|\right] \\
&\leq \frac{1}{k}\sum_{i=1}^{k}\epsilon(m) = \epsilon(m)
\end{aligned}$$

Where,

$$\epsilon(m) := \frac{4 + \sqrt{\log(\sup_B \tau_B(2m)) + \log(\tau_K(2m))}}{\sqrt{2m}}$$

Therefore, we have,

$$\begin{aligned}
&\mathbb{E}_{O\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|E(B,K) - E_O(B,K)\right|\right] \leq \mathbb{E}_{O\sim\mathcal{D}[k,m],U\sim\mathcal{D}[k]}\left[\sup_{B\in\mathscr{C}}\left|E_U(B,K) - E_O(B,K)\right|\right] \\
&\leq \mathbb{E}_{O\sim\mathcal{D}[2k,m],U\sim\mathcal{D}[k]}\left[\sup_{B\in\mathscr{C}}\mathbb{E}_{O'}\left[\left|E_U(B,K) - E_{O'}(B,K) + E_{O'}(B,K) - E_O(B,K)\right||U\right]\right] \\
&\leq \mathbb{E}_{O\sim\mathcal{D}[k,m],U\sim\mathcal{D}[k]}\left[\sup_{B\in\mathscr{C}}\mathbb{E}_{O'}\left[\left|E_U(B,K) - E_{O'}(B,K)\right||U\right] + \mathbb{E}_{O'}\left[\left|E_{O'}(B,K) - E_O(B,K)\right||U\right]\right] \\
&\leq \mathbb{E}_{O\sim\mathcal{D}[k,m],U\sim\mathcal{D}[k]}\left[\sup_{B\in\mathscr{C}}\epsilon(m) + \mathbb{E}_{O'}\left[\left|E_{O'}(B,K) - E_O(B,K)\right||U\right]\right] \\
&\leq \mathbb{E}_{O,O'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}E_{o_i}(B,K) - E_{o'_i}(B,K)\right|\right] + \epsilon(m)
\end{aligned}$$

We consider that $o_1, ..., o_k, o'_1, ..., o'_k$ are i.i.d samples of $\mathcal{D}[1, m]$ and denote:

$$\mu_i = \mathrm{err}_{o_i}(B,K) - \mathrm{err}_{o'_i}(B,K)$$

Reformulating the expression,

$$\mathbb{E}_{O,O'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}\mathrm{err}_{o_i}(B,K) - \mathrm{err}_{o'_i}(B,K)\right|\right] = \mathbb{E}_{O,O'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}\mu_i\right|\right]$$

Since $o_1, ..., o_k, o'_1, ..., o'_k$ are i.i.d samples replacing any μ_i with $-\mu_i$ will not affect the above expected value. In general, for any vector $\sigma \in \{\pm 1\}^k$ we have,

$$\mathbb{E}_{O,O'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}\mu_i\right|\right] = \mathbb{E}_{O,O'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}\sigma_i\mu_i\right|\right]$$

In particular, we can take expectation over σ that is sampled uniformly

$$\mathbb{E}_{\sigma}\mathbb{E}_{O,O'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}\sigma_i\mu_i\right|\right] = \mathbb{E}_{O,O'\sim\mathcal{D}[2k,m]}\mathbb{E}_{\sigma}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}\sigma_i\mu_i\right|\right]$$

For any fixed $\Lambda = O \cup O'$, we can take supermum only over the configurations

$$C = (c_{1,1}, c_{1,2}, ..., c_{k,1}, c_{k,2}, \bar{c}_{1,1}, \bar{c}_{1,2}, ..., \bar{c}_{k,1}, \bar{c}_{k,2}) \in [\mathcal{H}, \mathscr{C}, K]_\Lambda$$

It an also be said that,

$$\mathbb{E}_\sigma\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^{k}\sigma_i\mu_i\right|\right] = \mathbb{E}_\sigma\left[\max_{C\in[\mathcal{H},\mathscr{C},K]_\Lambda}\left|\frac{1}{k}\sum_{i=1}^{k}\sigma_i[\epsilon_{o_i}(c_{i,1},c_{i,2})-\epsilon_{o_i'}(\bar{c}_{i,1},\bar{c}_{i,2})]\right|\right]$$

For any C, we denote $\theta_C = \frac{1}{k}\sum_{i=1}^{k}\sigma_i[\epsilon_{o_i}(c_{i,1},c_{i,2})-\epsilon_{o_i'}(\bar{c}_{i,1},\bar{c}_{i,2})]$. We have $\mathbb{E}[\theta_C]=0$ and $\theta_C\in[-1,1]$. Therefore, by Hoeffding's inequality for all ρ,

$$\mathbb{P}[|\theta_C|>\rho]\le 2\exp\left(-2k\rho^2\right)$$

And by union bound over all $C\in[\mathcal{H},\mathscr{C},K]_\Lambda$,

$$\mathbb{P}\left[\max_{C\in[\mathcal{H},\mathscr{C},K]_\Lambda}|\theta_C|>\rho\right]\le 2|\,[\mathcal{H},\mathscr{C},K]_\Lambda\,|\exp\left(-2k\rho^2\right)\le 2\tau(2k,m)\exp\left(-2k\rho^2\right)$$

That yields:

$$\mathbb{E}_\sigma\left[\max_{C\in[\mathcal{H},\mathscr{C},K]_\Lambda}|\theta_C|\right]\le\frac{4+\sqrt{\log(\tau(2k,m;\mathscr{C},K))}}{\sqrt{2k}}$$

Therefore,

$$\mathbb{E}_{O,O'}\mathbb{E}_\sigma\left[\max_{C\in[\mathcal{H},\mathscr{C},K]_\Lambda}|\theta_C|\right]\le\frac{4+\sqrt{\log(\tau(2k,m;\mathscr{C},K))}}{\sqrt{2k}}$$

Combining the results,

$$\forall\mathcal{D}\ \ \forall K\in\mathscr{C}:\mathbb{E}_{O\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|E(B,K)-E_O(B,K)\right|\right]\le\frac{4+\sqrt{\log(\tau(2k,m;\mathscr{C},K))}}{\sqrt{2k}}+\epsilon(m)$$

□

In the proof of Theorem 5, we combine two bounds. The first bound is Equation 10 and the second is 11.

$$\forall p:\mathbb{P}_{o\sim p^m}\left[\forall c_1,c_2\in\mathcal{H}:\epsilon_{p_i}(c_1,c_2)\le\epsilon_o(c_1,c_2)+\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(4/\delta)}{m}}+\frac{1}{m}\right]\ge 1-\delta \quad (10)$$

Proof. This is an immediate extension of Theorem (cf. Vapnik (1998), Page 130). We only explain the modifications in the proof.

The difference between the bounds is that one handles only one arbitrary concept $c\in\mathcal{H}$ while the second handles two arbitrary concepts $c_1,c_2\in\mathcal{H}$. In the original proof, for a given unlabeled data set $o\cup o'$ of size $2m$ the concepts are separated into equivalence classes by their labelings on $o\cup o'$. In addition, the number of such labelings is count and is bounded by $\tau_{\mathcal{H}}(2m)$.

In our case, since the arbitrary selection of c_1,c_2, instead of counting the configurations of labelings over $o\cup o'$, we count $c_1\in\mathcal{H}_{o\cup o'}$ and $c_2\in\mathcal{H}_{o\cup o'}$ instead. Therefore, the number of labeling becomes $\tau_{\mathcal{H}}(2m)^2$ and $\log(\tau_{\mathcal{H}}(2m))$ is replaced with $2\log(\tau_{\mathcal{H}}(2m))$. □

The second bound states that: for all $\delta\in(0,1)$ with probability $\ge 1-\delta$ over i.i.d choice of $U=\{p_1,...,p_k\}\sim\mathcal{D}[k]$ we have,

$$\forall Q\in\mathcal{Q}:\quad E(Q,K)\le\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}[\mathrm{err}_{p_i}(B,K)]+\sqrt{\frac{\mathrm{KL}(Q\|P)+\log(k/\delta)}{2(k-1)}} \quad (11)$$

Proof. Let $\mathcal{H}=\bigcup_{B\in\mathscr{C}}B$ be a hypothesis class. In addition, P a prior distribution and $\mathcal{Q}$ a family of posterior distributions, both over $\mathscr{C}$. We denote $\mathcal{E}'$ the set of all distributions over $\mathcal{X}$ and objective function $G:\mathscr{C}\times\mathcal{E}\to[0,1]$ defined as $G(B,p)=\mathrm{err}_p(B,K)$. Applying Theorem 2 with examples set $\mathcal{E}'$, hypothesis class $\mathscr{C}$ and objective function G. The distribution is $\mathcal{D}$ over $\mathcal{E}'$. □

Theorem 5. Using Equation 11 with parameter $\delta/2$,

$$\mathbb{P}_{U\sim\mathcal{D}[k]}\left[\forall Q\in\mathcal{Q}: E(Q,K)\leq\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}[\mathrm{err}_{p_i}(B,K)]+\sqrt{\frac{\mathrm{KL}(Q\|P)+\log(2k/\delta)}{2(k-1)}}\right]\geq 1-\delta/2$$

The bound still holds if samples are selected according to each p_i along with the selection of U. Alternatively said, if $o_i\sim p_i^m$ and $U=\{p_1,...,p_k\}\sim\mathcal{D}[k]$:

$$\mathbb{P}_{O\sim\mathcal{D}[k,m]}\left[\forall Q\in\mathcal{Q}: E(Q,K)\leq\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}[\mathrm{err}_{p_i}(B,K)]+\sqrt{\frac{\mathrm{KL}(Q\|P)+\log(2k/\delta)}{2(k-1)}}\right]\geq 1-\delta/2 \tag{12}$$

By Equation 10, for each $p_i\in U$,

$$\mathbb{P}_{o_i\sim p_i^m}\left[\forall c_1,c_2\in\mathcal{H}:\epsilon_{p_i}(c_1,c_2)\leq\epsilon_{o_i}(c_1,c_2)+\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}\right]\geq 1-\lambda\delta/2$$

In particular, with probability $\geq 1-\lambda\delta/2$ over $o_i\sim p_i^m$:

$$\forall B,K\in\mathscr{C},\ \forall c_2\in K,\ c_1\in B:\epsilon_{p_i}(c_1,c_2)\leq\epsilon_{o_i}(c_1,c_2)+\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}$$

Alternatively:

$$\forall B\in\mathscr{C}:\mathrm{err}_{p_i}(B,K)\leq\mathrm{err}_{o_i}(B,K)+\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}$$

Next, we take expectation in both sides with respect to different Q,

$$\mathbb{P}_{o_i\sim d_i^m}\left[\forall Q\in\mathcal{Q}:\mathbb{E}_{B\sim Q}\left[\mathrm{err}_{p_i}(B,K)\right]\leq\mathbb{E}_{B\sim Q}\left[\mathrm{err}_{o_i}(B,K)\right]+\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}\right]\geq 1-\lambda\delta/2 \tag{13}$$

We define random variables $\{X_i\}_{i=1}^k$, each indicates if the i'th bound Equation. 13 holds uniformly (returns 0 if holds and 1 elsewise). This is a list of k independent random variables between 0 and 1. We denote $X=\frac{1}{k}\sum_{i=1}^k X_i$ and by Hoeffding's inequality,

$$\mathbb{P}\left[X\leq t+\mathbb{E}[X]\leq t+\frac{\lambda\delta}{2}\right]\geq 1-\exp(-2kt^2)\geq 1-\delta/2$$

We select $t=\lambda\delta/2$ and the inequality $1-\exp(-2kt^2)\geq 1-\delta/2$ holds whenever $k\geq\frac{8\log\left(\frac{2}{\delta}\right)}{(\lambda\delta)^2}$. It provides that $\mathbb{P}[X\leq\lambda\delta]\geq 1-\delta/2$. Thus, (with probability at least $1-\delta/2$) at least $1-\lambda\delta$ of the bounds hold uniformly for all Q. Any other bound, indexed i is then replaced with the bound $\mathbb{E}_{B\sim Q}[\mathrm{err}_{p_i}(B,K)]\leq 1+\mathbb{E}_{B\sim Q}[\mathrm{err}_{o_i}(B,K)]$ that holds for all Q with probability 1. The sum of the bounds is at most,

$$\lambda\delta k+\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}[\mathrm{err}_{o_i}(B,K)]+k\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{k}{m}$$

That bounds $\sum_{i=1}^k\mathbb{E}_{B\sim Q}[\mathrm{err}_{p_i}(B,K)]$ for all Q with probability at least $1-\delta/2$.

Alternatively, for all $U=\{p_1,...,p_k\}$, with probability at least $1-\delta/2$ over $o_1\sim p_1^m,...,o_k\sim p_k^m$,

$$\forall Q\in\mathcal{Q}:\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}[\mathrm{err}_{p_i}(B,K)]\leq\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}\left[\mathrm{err}_{o_i}(B,K)\right]+\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}+\lambda\delta$$

In particular, for randomly selected $U=\{p_1,...,p_k\}$ with probability at least $1-\delta/2$,

$$\begin{aligned}\forall Q\in\mathcal{Q}:\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}[\mathrm{err}_{p_i}(B,K)]&\leq\frac{1}{k}\sum_{i=1}^{k}\mathbb{E}_{B\sim Q}\left[\mathrm{err}_{o_i}(B,K)\right]\\&+\sqrt{\frac{2\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}+\lambda\delta\end{aligned} \tag{14}$$

By union bound for Equation. 12 and Equation. 14, with probability at least $1-\delta$ (over $O\sim\mathcal{D}[k,m]$) we have the desired bound.

□

C Proofs for the claims in Section 5

Lemma 4. We suggest the following procedure for transferring the hypothesis class.

Input: $S = s_{[1,k]} \sim \mathcal{D}[k, m]$.
Output: hypothesis class $N(S)$ such that $\inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \epsilon$ with probability $\geq 1 - \delta$.
Narrowing apply the following procedure.

Case a If sampled at least 2 positive samples that are non co-linear with the point 0, return $N(S) =$ set of all disks on the sub-space of the three points (including $c \equiv 0$).

Case b If sampled at least 1 positive sample (x, y) only on the same line with 0, return $N(S) =$ set of all symmetric intervals on this line (including $c \equiv 0$).

Case c Else, return $N(S) = \{c \equiv 0\}$.

First, the VC dimension of any output subclass in the narrowing step is ≤ 1. Let A_ϵ be the set of tasks with positive a ratio $\leq \epsilon$ (i.e, probability $\leq \epsilon$ to draw a positive sample). We are left to show that the output $N(S)$ satisfies $\inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \epsilon$ with probability $\geq 1 - \delta$.

If $prob_D(A_\epsilon) \leq 1 - \delta$**;** the probability of sampling at least one task with positive ratio $\geq \epsilon$ is at least $(1 - (1 - \delta)^k)$. In addition, the probability of sampling at least 2 positive samples from that task is at least $1 - \sum_{i=0}^{1} \binom{m}{i} \cdot \epsilon^i (1 - \epsilon)^{m-i} \geq 1 - 2m \cdot (1 - \epsilon)^{m-1}$. If we choose $k > \frac{\log(\delta')}{1-\delta}$, we achieve $(1 - (1 - \delta)^k) > 1 - \delta'$ and when m as above, we get $1 - 2m \cdot (1 - \epsilon)^{m-1} > 1 - \delta'$. Therefore, with probability $\geq (1 - \delta')^2 = 1 - \delta$, the algorithm will return case a or case b. With additional similarconsiderations, we could distinguish between cases a and b. If case a is satisfied, the returned subclass, $N(S)$, achieves $\inf_{c \in N(S)} \epsilon_{d_t}(c) = 0 < \epsilon$.

Otherwise, in the case where $prob_D(A_\epsilon) \geq 1 - \delta$**;** then with probability $\geq 1 - \delta$ the target task is selected with positive ratio $\leq \epsilon$. Thus, $c \equiv 0$ satisfies $\epsilon_{d_t}(c) \leq \epsilon$ and in each case a,b or c, we achieve $\inf_{c \in N(S)} \epsilon_{d_t}(c) \leq \epsilon$ since $c \equiv 0$ is always in the output bias.

□

We recall the definition of ϵ-representativeness of a data set of samples.

Definition 10. *$s = \{z_1, ..., z_m\}$ is ϵ-representative (w.r.t $Z, \mathcal{H}, \ell, d$) if,*

$$\forall c \in \mathcal{H} : \quad \left| \epsilon_s(c) - \epsilon_d(c) \right| \leq \epsilon$$

Before proving Lemma 5, we introduce and prove the following.

Lemma 9. *Let $S = s_{[1,k]}$ be a source data set. Assume that each data set s_i is $\frac{\epsilon}{2}$-representative (w.r.t $Z, \mathcal{H}, \ell, d_i$). Then we have, $\epsilon_U(F) \leq \inf_{B \in \mathscr{C}} \epsilon_U(B) + \epsilon$ for any $F =$ C-ERM$_{\mathscr{C}}(S)$.*

Proof. Take any $i \in [k]$. For every $c \in B$ we have,

$$\epsilon_{s_i}(c^*_{i,B}) \leq \epsilon_{s_i}(c) \leq \epsilon_{d_i}(c) + \frac{\epsilon}{2}$$

Thus, $\forall i \in [k]$ we have: $\forall B \in \mathscr{C} : \quad \epsilon_{s_i}(c^*_{i,B}) \leq \inf_{c \in B} \epsilon_{d_i}(c) + \frac{\epsilon}{2}$. We conclude,

$$\min_{B \in \mathscr{C}} \frac{1}{k} \sum_{i=1}^{k} \epsilon_{s_i}(c^*_{i,B}) \leq \inf_{B \in \mathscr{C}} \frac{1}{k} \sum_{i=1}^{k} \inf_{c \in B} \epsilon_{d_i}(c) + \frac{\epsilon}{2}$$

It can also be said that if $F =$ C-ERM$_{\mathscr{C}}(S) = \arg\min_{B \in \mathscr{C}} \frac{1}{k} \sum_{i=1}^{k} \epsilon_{s_i}(c^*_{i,B})$ then its empirical transfer risk is:

$$\epsilon_U(F) = \frac{1}{k} \sum_{i=1}^{k} \inf_{c \in F} \epsilon_{d_i}(c) \leq \frac{1}{k} \sum_{i=1}^{k} \epsilon_{d_i}(c^*_{i,F}) \leq \frac{1}{k} \sum_{i=1}^{k} \epsilon_{s_i}(c^*_{i,F}) + \frac{\epsilon}{2}$$

$$\leq \inf_{B \in \mathscr{C}} \frac{1}{k} \sum_{i=1}^{k} \epsilon_{d_i}(c^*_{i,B}) + \frac{\epsilon}{2} + \frac{\epsilon}{2} \leq \inf_{B \in \mathscr{C}} \epsilon_U(B) + \epsilon$$

□

Lemma 5. By the uniform convergence property of $\mathscr{C}$, for any ϵ, δ for any k larger than some function of ϵ, δ,

$$\mathbb{P}_{U\sim\mathcal{D}[k]}\left[\forall B : \left|\epsilon_{\mathcal{D}}(B) - \epsilon_U(B)\right| \leq \frac{\epsilon}{2}\right] \geq 1 - \delta/2 \tag{15}$$

In addition, for any $\epsilon, \delta, i \in [k]$ there is m for which $s_i \sim d_i^m$ is $\frac{\epsilon}{4}$-representative with probability $\geq 1 - \delta/2k$. By union bound, $s_i \sim d_i^m$ are all $\frac{\epsilon}{4}$-representative with probability $\geq 1 - \delta/2$.

Thus, by Lemma 9 we have,

$$\mathbb{P}_{S\sim\mathcal{D}[k,m]}\left[\epsilon_U(F) \leq \inf_{B\in\mathscr{C}} \epsilon_U(B) + \frac{\epsilon}{2}\right] \geq 1 - \delta/2 \tag{16}$$

Where $F = \text{C-ERM}_{\mathscr{C}}(S)$. Again, by applying union bound on Equation 15 and Equation 16 we have:

$$\mathbb{P}_{S\sim\mathcal{D}[k,m]}\left[\epsilon_{\mathcal{D}}(F) \leq \inf_{B\in\mathscr{C}} \epsilon_U(B) + \frac{\epsilon}{2} + \frac{\epsilon}{2} = \inf_{B\in\mathscr{C}} \epsilon_{\mathcal{D}}(B) + \epsilon\right] \geq 1 - \delta \tag{17}$$

Which concludes the proof of this Lemma. □

Lemma 6. The proof of this lemma is divided into two parts. The first part shows that there might be a simplifier even if the hypothesis class is unlearnable. The second part shows that the complexity of the outputs is unbounded.

Part 1. We define $\mathscr{C}$ as follows,

$$\mathscr{C} = \{\{c \equiv 0\}\} \cup \{\mathcal{H}_n : n \in \mathbb{N}\}$$

Where, $\mathcal{H}_n$ is any hypothesis class of VC dimension n consisting of functions of the form:

$$c : [0,\infty) \times [0,1] \to \{0,1\}, \quad c(x) = 0 \ \ \forall x \in \{[0,\infty) \setminus [n-1,n)\} \times [0,1]$$

The union $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$ has VC dimension $\geq n$ for all $n \in \mathbb{N}$. Therefore, it is infinite and by the fundamental Theorem of learnability Vapnik & Chervonenkis (1971),Blumer et al. (1989), the concept class $\mathcal{H}$ is not learnable. We mention that $\forall c \in \mathcal{H}_n : c(x) = 1 \implies x \in [n-1,n) \times [0,1]$. Another way saying it is if a positive sample $(x, 1)$ is sampled then one can identify n very easily. The following procedure, N, behaves as a simplifier,

Input: $S \sim \mathcal{D}[k,m]$.
Output: hypothesis class $N(S)$ such that $\inf_{c\in N(S)} \epsilon_{d_t}(c) \leq \epsilon$ with probability $\geq 1 - \delta$.
Narrowing apply the following procedure.

Case a If at least one positive example $z = (x, 1)$ is drawn, for x that lies in $[n-1,n) \times [0,1]$ (for some $n \in \mathbb{N}$), return $\mathcal{H}_n$.

Case b Otherwise, return $\{c \equiv 0\}$.

Let A_ϵ be the set of tasks with ratio $\leq \epsilon$ of positive examples (i.e, probability $\leq \epsilon$ to draw a positive example). We will show that the output $N(S)$ satisfies $\inf_{c\in N(S)} \epsilon_d(c) \leq \epsilon$ with probability $\geq 1 - \delta$ for a random task, d.

If $prob_{\mathcal{D}}(A_\epsilon) \geq 1 - \delta$**;** then with probability $\geq 1 - \delta$, the target task is selected with positive ratio $\leq \epsilon$. Thus, $c \equiv 0$ satisfies $\epsilon_d(c) \leq \epsilon$ and in case a we always have $\inf_{c\in N(S)} \epsilon_d(c) = 0 \leq \epsilon$.

Otherwise in the case when $prob_{\mathcal{D}}(A_\epsilon) \leq 1-\delta$**;** the probability of sampling at least one task with positive ratio $\geq \epsilon$ is at least $1 - (1-\delta)^k$. In addition, the probability of sampling at least one positive sample from that task is at least $1-(1-\epsilon)^m$. We can choose k that achieves $1-(1-\delta)^k > \sqrt{1-\delta}$ and when m is large enough (as a function of both ϵ, δ), we have $1 - (1-\epsilon)^m > \sqrt{1-\delta}$. Therefore, with probability $\geq (\sqrt{1-\delta})^2 = 1 - \delta$, the algorithm will return case a. i.e, a subclass $N(S)$ that achieves $\inf_{c\in N(S)} \epsilon_d(c) = 0 < \epsilon$.

Part 2. Let $\mathcal{H} = \bigcup_{B \in \mathscr{C}} B$ be a hypothesis class of infinite VC dimension and $\max_{B \in \mathscr{C}} \mathrm{vc}(B) = v < \infty$. Assume that there is a simplifier A. By Lemma 1, the sample complexity for learning with ERM any hypothesis class $B \in \mathscr{C}$ is bounded by a universal function of (v, ϵ, δ) that will be denoted by $N(\epsilon, \delta)$ that does not depend on B. We construct a learner for $\mathcal{H}$ as follows,

Input: $S \sim d^{mk+n}$.
Output: concept c_{out} such that $\epsilon_{d_t}(c_{out}) \leq \epsilon$ with probability $\geq 1 - \delta$.
Partition the data $S = (s_{[1,k]}, s_t)$ (s_i the i'th consecutive m examples and s_t, last n examples).;
Narrowing simulate the simplifier $B := A(s_{[1,k]})$.
Output: $c_{out} = \mathrm{ERM}_B(s_t)$.

Here, k, m are the required complexities for the simplifier with error and confidence parameters $(\epsilon/2, \delta/2)$ and $n = N(\epsilon/2, \delta/2)$. We treat d as any distribution over Z.

By the definition of the simplifier, with probability $\geq 1 - \delta/2$, B satisfies $\epsilon_{\mathcal{D}}(B) \leq \inf_{X \in \mathscr{C}} \epsilon_{\mathcal{D}}(X) + \frac{\epsilon}{2}$ for any factory $\mathcal{D}$ with input $[k, m]$. In particular, for the factory that is supported only by d.

Alternatively,

$$\mathbb{P}_{s_{[1,k]} \sim d^{mk}} \left[\epsilon_{\mathcal{D}}(B) \leq \inf_{X \in \mathscr{C}} \epsilon_{\mathcal{D}}(X) + \frac{\epsilon}{2} \right] \geq 1 - \delta/2$$

The way we defined the factory yields that $\epsilon_{\mathcal{D}}(B) = \inf_{c \in B} \epsilon_d(c)$. Thus,

$$\mathbb{P}_{s_{[1,k]} \sim d^{mk}} \left[\inf_{c \in B} \epsilon_d(c) \leq \inf_{c \in \mathcal{H}} \epsilon_d(c) + \frac{\epsilon}{2} \right] \geq 1 - \delta/2$$

Furthermore, since ERM for any B returns a concept that has error at most $\epsilon/2$ with probability $\geq 1 - \delta/2$ for $n = N(\epsilon/2, \delta/2)$ samples,

$$\mathbb{P}_{s_t \sim d^n} \left[\epsilon_d(c_{out}) \leq \inf_{c \in B} \epsilon_d(c) + \epsilon/2 \right] \geq 1 - \delta/2$$

By union bound, we have the desired,

$$\mathbb{P}_{S \sim d^{mk+n}} \left[\epsilon_d(c_{out}) \leq \epsilon \right] \geq 1 - \delta$$

□

D Proofs for the claims in Section 5.2

Theorem 7. First, we note that $\mathbb{E}_U[\epsilon_U(B)] = \epsilon_{\mathcal{D}}(B)$. Thus, we have,

$$\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{\mathcal{D}}(B)-\epsilon_S(B,r)\right|\right]=\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\mathbb{E}_{U\sim\mathcal{D}^k}[\epsilon_U(B)-\epsilon_S(B,r)]\right|\right]$$

By the triangle inequality and the fact that $\sup_c \mathbb{E}[X_c] \le \mathbb{E}[\sup_c X_c]$ we have,

$$\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{\mathcal{D}}(B)-\epsilon_S(B,r)\right|\right]\le\mathbb{E}_{S,U}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_U(B)-\epsilon_S(B,r)\right|\right]$$

Let $S' = (s'_1, ..., s'_k)$ be $s'_i \sim d_i^m$ for all $i \in [k]$ where $U = \{d_1, .., d_k\}$, then by the triangle inequality,

$$\begin{aligned}\mathbb{E}_{S'}\left[\left|\epsilon_U(B)-\epsilon_{S'}(B,r)\right|\right]&=\mathbb{E}_{S'}\left[\left|\frac{1}{k}\sum_{i=1}^k\epsilon_{d_i}(B)-\epsilon_{s'_i}(r_B(s'_i))\right|\right]\\&\le\frac{1}{k}\sum_{i=1}^k\mathbb{E}_{s'_i}\left[\left|\epsilon_{d_i}(B)-\epsilon_{s'_i}(r_B(s'_i))\right|\right]\le\frac{1}{k}\sum_{i=1}^k\epsilon(m)=\epsilon(m)\end{aligned}$$

Therefore, for $S' = (s'_1, ..., s'_k)$ such that $s'_i \sim d_i^m$ (for all $i \in [k]$) where $U = \{d_1, .., d_k\} \sim \mathcal{D}[k]$ and $S \sim \mathcal{D}[k, m]$ and the fact that $\sup_c \mathbb{E}[X_c] \le \mathbb{E}[\sup_c X_c]$ we have,

$$\begin{aligned}\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{\mathcal{D}}(B)-\epsilon_S(B,r)\right|\right]&\le\mathbb{E}_{S,U}\left[\sup_{B\in\mathscr{C}}\mathbb{E}_{S'}\left[\left|\epsilon_U(B)-\epsilon_{S'}(B,r)+\epsilon_{S'}(B,r)-\epsilon_S(B,r)\right|\right]\right]\\&\le\mathbb{E}_{S,U}\left[\sup_{B\in\mathscr{C}}\mathbb{E}_{S'}\left[\left|\epsilon_{S'}(B,r)-\epsilon_S(B,r)\right|\right]+\epsilon(m)\right]\\&\le\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{S'}(B)-\epsilon_S(B,r)\right|\right]+\epsilon(m)\\&=\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\epsilon_{s_i}(r_B(s_i))-\epsilon_{s'_i}(r_B(s'_i))\right|\right]+\epsilon(m)\end{aligned}$$

We consider that $s_1, ..., s_k, s'_1, ..., s'_k$ are i.i.d samples of $\mathcal{D}[1, m]$ and denote:

$$\mu_i = \epsilon_{s_i}(r_B(s_i)) - \epsilon_{s'_i}(r_B(s'_i))$$

Reformulating the expression,

$$\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\epsilon_{s_i}(r_B(s_i))-\epsilon_{s'_i}(r_B(s'_i))\right|\right]=\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\mu_i\right|\right]$$

Since $s_1, ..., s_k, s'_1, ..., s'_k$ are i.i.d samples replacing any μ_i with $-\mu_i$ will not affect the above expected value. In general, for any vector $\sigma \in \{\pm 1\}^k$ we have,

$$\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\mu_i\right|\right]=\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\sigma_i\mu_i\right|\right]$$

In particular, we can take expectation over σ that is sampled uniformly

$$\mathbb{E}_\sigma\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\sigma_i\mu_i\right|\right]=\mathbb{E}_{S,S'\sim\mathcal{D}[2k,m]}\mathbb{E}_\sigma\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\sigma_i\mu_i\right|\right]$$

For any fixed $\Lambda = S \cup S'$, we can take supermum only over the configurations

$$C = (c_1, ..., c_k, \bar{c}_1, ..., \bar{c}_k) \in [\mathcal{H}, \mathscr{C}, r]_\Lambda$$

It can also be stated that,

$$\mathbb{E}_\sigma\left[\sup_{B\in\mathscr{C}}\left|\frac{1}{k}\sum_{i=1}^k\sigma_i\mu_i\right|\right]=\mathbb{E}_\sigma\left[\max_{C\in[\mathcal{H},\mathscr{C},r]_\Lambda}\left|\frac{1}{k}\sum_{i=1}^k\sigma_i[\epsilon_{s_i}(c_i)-\epsilon_{s'_i}(\bar{c}_i)]\right|\right]$$

For any C, we denote $\theta_C = \frac{1}{k}\sum_{i=1}^{k}\sigma_i[\epsilon_{s_i}(c_i) - \epsilon_{s'_i}(\bar{c}_i)]$. We have $\mathbb{E}[\theta_C] = 0$ and $\theta_C \in [-1, 1]$. Therefore, by Hoeffding's inequality for all ρ,

$$\mathbb{P}[|\theta_C| > \rho] \leq 2\exp\left(-2k\rho^2\right)$$

By union bound over all $C \in [\mathcal{H}, \mathscr{C}, r]_\Lambda$,

$$\mathbb{P}\left[\max_{C\in[\mathcal{H},\mathscr{C},r]_\Lambda} |\theta_C| > \rho\right] \leq 2|\,[\mathcal{H}, \mathscr{C}, r]_\Lambda\,|\exp\left(-2k\rho^2\right) \leq 2\tau(2k, m, r)\exp\left(-2k\rho^2\right)$$

That yields:

$$\mathbb{E}_\sigma\left[\max_{C\in[\mathcal{H},\mathscr{C},r]_\Lambda} |\theta_C|\right] \leq \frac{4 + \sqrt{\log(\tau(2k, m, r))}}{\sqrt{2k}}$$

Therefore,

$$\mathbb{E}_{S,S'}\mathbb{E}_\sigma\left[\max_{C\in[\mathcal{H},\mathscr{C},r]_\Lambda} |\theta_C|\right] \leq \frac{4 + \sqrt{\log(\tau(2k, m, r))}}{\sqrt{2k}}$$

Combining the results,

$$\mathbb{E}_{S\sim\mathcal{D}[k,m]}\left[\sup_{B\in\mathscr{C}}\left|\epsilon_{\mathcal{D}}(B) - \epsilon_S(B, r)\right|\right] \leq \frac{4 + \sqrt{\log(\tau(2k, m, r))}}{\sqrt{2k}} + \epsilon(m)$$

□

Let $\mathcal{H} = \bigcup_{B\in\mathscr{C}} B$ be a hypothesis class. In addition, P a prior distribution and $\mathcal{Q}$ a family of posterior distributions, both over $\mathscr{C}$. Applying Theorem 2 with examples set $\mathcal{E}$, hypothesis class $\mathscr{C}$ and objective function $g : \mathscr{C} \times \mathcal{E} \to [0, 1]$. In addition, the distribution is $\mathcal{D}$ over $\mathcal{E}$. Thus, for all $\delta \in (0, 1)$ with probability $\geq 1 - \delta$ over i.i.d choice of $U = \{d_1, ..., d_k\} \sim \mathcal{D}[k]$:

$$\forall Q \in \mathcal{Q}: \quad R(Q) \leq R_U(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(k/\delta)}{2(k-1)}} \tag{18}$$

Theorem 8. Using Equation. 18 with parameter $\delta/2$,

$$\mathbb{P}_{U\sim\mathcal{D}[k]}\left[\forall Q \in \mathcal{Q} : R(Q) \leq R_U(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}}\right] \geq 1 - \delta/2$$

The bound still holds if samples are selected according to each d_i along with the selection of U. Another way of saying it is, if $s_i \sim d_i^m$ and $U = \{d_1, ..., d_k\} \sim \mathcal{D}[k]$:

$$\mathbb{P}_{S\sim\mathcal{D}[k,m]}\left[\forall Q \in \mathcal{Q} : R(Q) \leq R_U(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}}\right] \geq 1 - \delta/2 \tag{19}$$

For each $d_i \in U$ we have,

$$\mathbb{P}_{s_i\sim d_i^m}\left[\forall c \in \mathcal{H} : \epsilon_{d_i}(c) \leq \epsilon_{s_i}(c) + \sqrt{\frac{\log(\tau_{\mathcal{H}}(2m)) + \log(8/\lambda\delta)}{m}} + \frac{1}{m}\right] \geq 1 - \lambda\delta/2$$

By Equation 4.26 in (cf. Vapnik (1998), Page 130). In particular,

$$\mathbb{P}_{s_i\sim d_i^m}\left[\forall B \in \mathscr{C},\ \forall c \in B : \epsilon_{d_i}(c) \leq \epsilon_{s_i}(c) + \sqrt{\frac{\log(\tau_{\mathcal{H}}(2m)) + \log(8/\lambda\delta)}{m}} + \frac{1}{m}\right] \geq 1 - \lambda\delta/2$$

In addition,

$$\mathbb{P}_{s_i\sim d_i^m}\left[\forall B \in \mathscr{C} : \inf_{c\in B}\epsilon_{d_i}(c) \leq \epsilon_{d_i}(c^*_{i,B}) \leq \epsilon_{s_i}(c^*_{i,B}) + \sqrt{\frac{\log(\tau_{\mathcal{H}}(2m)) + \log(8/\lambda\delta)}{m}} + \frac{1}{m}\right] \geq 1 - \lambda\delta/2$$

Next, we take expectation in both sides with respect to different Q,

$$\mathbb{P}_{s_i\sim d_i^m}\left[\forall Q\in\mathcal{Q}:\mathbb{E}_{B\sim Q}\left[\inf_{c\in B}\epsilon_{d_i}(c)\right]\leq\mathbb{E}_{B\sim Q}\left[\epsilon_{s_i}(c^*_{i,B})\right]+\sqrt{\frac{\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}\right]\geq 1-\lambda\delta/2 \tag{20}$$

We define random variables $\{X_i\}_{i=1}^k$, each indicates if the i'th bound Equation. 20 holds uniformly (returns 0 if holds and 1 else). This is a list of k independent random variables between 0 and 1. We denote $X=\frac{1}{k}\sum_{i=1}^k X_i$ and by Hoeffding's inequality,

$$\mathbb{P}\left[X\leq t+\mathbb{E}[X]\leq t+\frac{\lambda\delta}{2}\right]\geq 1-\exp(-2kt^2)\geq 1-\delta/2$$

We select $t=\lambda\delta/2$ and the inequality $1-\exp(-2kt^2)\geq 1-\delta/2$ holds whenever $k\geq\frac{8\log\left(\frac{2}{\delta}\right)}{(\lambda\delta)^2}$. It provides that $\mathbb{P}\left[X\leq\lambda\delta\right]\geq 1-\delta/2$. Thus, (with probability at least $1-\delta/2$) at least $1-\lambda\delta$ of the bounds hold uniformly for all Q. For any other bound, indexed i is then replaced with the bound $\mathbb{E}_{B\sim Q}[\inf_{c\in B}\epsilon_{d_i}(c)]\leq 1+\mathbb{E}_{B\sim Q}[\epsilon_{s_i}(c^*_{i,B})]$ that holds for all Q with probability 1. The sum of the bounds is at most,

$$\lambda\delta k+\sum_{i=1}^k\mathbb{E}_{B\sim Q}[\epsilon_{s_i}(c^*_{i,B})]+k\sqrt{\frac{\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{k}{m}$$

That bounds $kR_U(Q)$ for all Q with probability at least $1-\delta/2$.

Alternatively, for all $d_1,...,d_k$ with probability $\geq 1-\delta/2$ over $s_i\sim d_i^m$ (for all $i\in[k]$):

$$\forall Q\in\mathcal{Q}:R_U(Q)\leq\frac{1}{k}\sum_{i=1}^k\mathbb{E}_{B\sim Q}\left[\epsilon_{s_i}(c^*_{i,B})\right]+\sqrt{\frac{\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}+\lambda\delta$$

In particular, for randomly selected $U=\{d_1,...,d_k\}$,

$$\mathbb{P}_{S\sim\mathcal{D}[k,m]}\left[\forall Q\in\mathcal{Q}:R_U(Q)\leq\frac{1}{k}\sum_{i=1}^k\mathbb{E}_{B\sim Q}\left[\epsilon_{s_i}(c^*_{i,B})\right]+\sqrt{\frac{\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}+\lambda\delta\right]\geq 1-\delta/2 \tag{21}$$

By union bound for Equation. 19 and Equation. 21, with probability at least $1-\delta$ (over $S\sim\mathcal{D}[k,m]$) we have,

$$\begin{aligned}\forall Q\in\mathcal{Q}:R(Q)\leq&\frac{1}{k}\sum_{i=1}^k\mathbb{E}_{B\sim Q}\left[\epsilon_{s_i}(c^*_{i,B})\right]\\&+\sqrt{\frac{\log(\tau_{\mathcal{H}}(2m))+\log(8/\lambda\delta)}{m}}+\frac{1}{m}+\sqrt{\frac{\mathrm{KL}(Q\|P)+\log(2k/\delta)}{2(k-1)}}+\lambda\delta\end{aligned}$$

□

Theorem 9. Following the same line of the proof of Theorem 8, we have,

$$\mathbb{P}_{S\sim\mathcal{D}[k,m]}\left[\forall Q\in\mathcal{Q}:R(Q)\leq R_U(Q)+\sqrt{\frac{\mathrm{KL}(Q\|P)+\log(2k/\delta)}{2(k-1)}}\right]\geq 1-\delta/2 \tag{22}$$

It is easy to verify that $\mathbb{E}_{B\sim Q}\left[\inf_{c\in B}\epsilon_{d_i}(c)\right]\leq\mathbb{E}_{c\sim Q_q}\left[\epsilon_{d_i}(c)\right]$. Hence by Theorem 2, for each $d_i\in U$, we have,

$$\mathbb{P}_{s_i\sim d_i^m}\left[\forall Q,q:\mathbb{E}_{B\sim Q}\left[\inf_{c\in B}\epsilon_{d_i}(c)\right]\leq\mathbb{E}_{c\sim Q_q}\left[\epsilon_{d_i}(c)\right]\leq\mathbb{E}_{c\sim Q_q}\left[\epsilon_{s_i}(c)\right]+\sqrt{\frac{\mathrm{KL}(Q_q\|p)+\log(2m/\lambda\delta)}{2(m-1)}}\right]\geq 1-\lambda\delta/2$$

Thus,

$$\mathbb{P}_{s_i\sim d_i^m}\left[\forall Q:\mathbb{E}_{B\sim Q}\left[\inf_{c\in B}\epsilon_{d_i}(c)\right]\leq\inf_q\mathbb{E}_{c\sim Q_q}[\epsilon_{s_i}(c)]+\sqrt{\frac{\mathrm{KL}(Q_q\|p)+\log(2m/\lambda\delta)}{2(m-1)}}\right]\geq 1-\lambda\delta/2 \tag{23}$$

Using the same method of Hoeffding's inequality from the proof of Theorem 8,

$$kR_U(Q) \leq \lambda\delta k + \sum_{i=1}^{k} \min_{q_i} \mathbb{E}_{c\sim Q_{q_i}}[\epsilon_{s_i}(c)] + k\sqrt{\frac{\mathrm{KL}(Q_{q_i}\|p) + \log(2m/\lambda\delta)}{2(m-1)}}$$

With probability at least $1 - \delta/2$. Alternatively, for all $d_1, ..., d_k$ with probability $\geq 1 - \delta/2$ over $s_i \sim d_i^m$ (for all $i \in [k]$):

$$\forall Q : R_U(Q) \leq \frac{1}{k}\sum_{i=1}^{k} \min_{q_i} \mathbb{E}_{c\sim Q_{q_i}}[\epsilon_{s_i}(c)] + \sqrt{\frac{\mathrm{KL}(Q_{q_i}\|p) + \log(2m/\lambda\delta)}{2(m-1)}} + \lambda\delta$$

In particular, for randomly selected $U = \{d_1, ..., d_k\}$,

$$\mathbb{P}_{S\sim\mathcal{D}[k,m]}\left[\forall Q : R_U(Q) \leq \frac{1}{k}\sum_{i=1}^{k} \min_{q_i} \mathbb{E}_{c\sim Q_{q_i}}[\epsilon_{s_i}(c)] + \sqrt{\frac{\mathrm{KL}(Q_{q_i}\|p) + \log(2m/\lambda\delta)}{2(m-1)}} + \lambda\delta\right] \geq 1 - \delta/2 \tag{24}$$

By union bound for Equation. 22 and Equation. 24, with probability at least $1 - \delta$ (on $S \sim \mathcal{D}[k, m]$), the following holds for all Q,

$$\begin{aligned} R(Q) \leq{}& \frac{1}{k}\sum_{i=1}^{k} \min_{q_i} \mathbb{E}_{c\sim Q_{q_i}}[\epsilon_{s_i}(c)] \\ &+ \sqrt{\frac{\mathrm{KL}(Q_{q_i}\|p) + \log(2m/\lambda\delta)}{2(m-1)}} + \sqrt{\frac{\mathrm{KL}(Q\|P) + \log(2k/\delta)}{2(k-1)}} + \lambda\delta \end{aligned}$$

□

Lemma 6. Recall the assumption,

$$r_{B_1}(s_1)(s_1) = r_{B_2}(s_2)(s_2) \text{ whenever } h_{B_1}(s_1) = h_{B_2}(s_2)$$

In particular, $r_{i,B_1}(s_i) = r_{i,B_2}(s_i)$ whenever $h_{B_1}(s_i) = h_{B_2}(s_i)$.

Therefore,

$$\begin{aligned} \left|\{r_{1,B}(s_1), ..., r_{k,B}(s_k) : B \in \mathscr{C}\}\right| &\leq \left|\left\{h_B(s_1), ..., h_B(s_k) : B \in \mathcal{H}^I_{V,E,\mathrm{sign}}\right\}\right| \\ &= \left|\left\{h_B(s_1 \cup ... \cup s_k) : B \in \mathcal{H}^I_{V,E,\mathrm{sign}}\right\}\right| \\ &= \left|\left\{h_B(\Lambda) : B \in \mathcal{H}^I_{V,E,\mathrm{sign}}\right\}\right| \\ &\leq \tau_t(|\Lambda|) = \tau_t(mk) \end{aligned}$$

Where $\Lambda = s_1 \cup ... \cup s_k$.

Thus,

$$\tau(k, m, r) \leq \tau_t(mk)$$

By analysis due to Kakade & Tewari (2008),

$$\tau(k, m, r) \leq \tau_t(mk) \leq (emk)^{|I|}$$

□

E Proofs for Section 6

Theorem 11. We show that the post transfer learning rule r is endowed with the required properties.

- **Equation 3:** Let $h_1 = \mathrm{ERM}_{\mathcal{H}_u}(h_{B_1}(s_1))$ and $h_2 = \mathrm{ERM}_{\mathcal{H}_u}(h_{B_2}(s_2))$. These are equal, since $h_{B_1}(s_1) = h_{B_2}(s_2)$. Therefore, $h_1 \circ h_{B_1}(s_1) = h_2 \circ h_{B_2}(s_2)$.
- **Equation 2:** For any fixed B we have that r_B is simply an ERM rule of the hypothesis class B that has growth function $\leq (em)^{|J|}$ since $\mathcal{H}_u$ is an architecture with $|J|$ parameters. Therefore by Theorem 1,

$$\mathbb{E}_{s\sim d^m}\left[\left|\epsilon_d(r(s)) - \epsilon_s(r(s))\right|\right] \leq \frac{4 + \sqrt{|J|\log(2em)}}{\sqrt{2m}}$$

That yields:

$$\mathbb{E}_{s\sim d^m}\left[\left|\epsilon_d(r(s)) - \inf_{c\in B}\epsilon_s(c)\right|\right] \leq \frac{4 + \sqrt{|J|\log(2em)}}{\sqrt{2m}}$$

Therefore,

$$\mathbb{E}_{s\sim d^m}\left[\left|\epsilon_s(r(s)) - \inf_{c\in B}\epsilon_s(c)\right|\right] \leq \frac{8 + \sqrt{4|J|\log(2em)}}{\sqrt{2m}}$$

Alternatively, $\epsilon(m) = \frac{8+\sqrt{4|J|\log(2em)}}{\sqrt{2m}}$.

□

Theorem 12. We use Theorem 8 and Equation 4. In addition, by analysis due to Kakade & Tewari (2008) we obtain $\tau_{\mathcal{H}}(m) \leq (em)^{|E|}$. □

Lemma 7. This is an extension of (cf. Shalev-Shwartz & Ben-David (2014), Theorem 20.6) which is based on Kakade & Tewari (2008). We will only explain the modifications in the proof.

We denote the growth function of the hypothesis class B by τ. In addition, $\tau_{t,i}(m)$ is the growth function of neuron i in layer t. It follows that,

$$\tau_{\mathcal{H}}(m) \leq \prod_{t,i} \tau_{t,i}(m)$$

Neuron i in layer t is a homogenous halfspace hypothesis with $d_{t,i}$ unfixed entries and $d'_{t,i}$ entries in total. We observe this is a hypothesis class,

$$Q = \{\mathrm{sign}(g(x)) : g \in \mathcal{G}\}$$

where $\mathcal{G}$ the set of all functions of the form $\langle w, x\rangle$ with w fixed in $d'_{t,i} - d_{t,i}$ specified indexes. This is a vector space of functions $g : \mathbb{R}^{d'_{t,i}} \to \mathbb{R}$ of dimension $d_{t,i}$. Thus, Q has VC dimension $\leq d_{t,i}$. By Equation 1, the growth function of this hypothesis class is $\tau_{t,i}(m) \leq (em)^{d_{t,i}}$.

The overall growth function becomes,

$$\tau(m) \leq (em)^{\sum_{t,i} d_{t,i}} = (em)^{|J|}$$

If the VC dimension is m then, $\tau(m) = 2^m$ and so we obtain $m \leq |J|\log(em)/\log(2)$ which yields the assymptotic behaviour above by Equation. 5. □

F Proofs for Section 6.3

Equation 6. By Theorem 10 with $\mathcal{H} = \mathcal{H}_{V,E,\text{sign}}$, $\mathscr{C} = \mathcal{H}^{E}_{V,E,\text{sign}}$ and $\mathcal{D}$ (as before) along to Markov's inequality, with probability $\geq 1 - \delta$ (over S),

$$\begin{aligned}
\forall c \in \mathcal{H} : \epsilon_{\mathcal{D}}(c_S) &\leq \epsilon_S(c_S, r) + \frac{4 + \sqrt{|E| \cdot \log(2emk)}}{\delta\sqrt{2k}} + \frac{8}{\delta\sqrt{2m}} \\
&\leq \epsilon_S(c, r) + \frac{4 + \sqrt{|E| \cdot \log(2emk)}}{\delta\sqrt{2k}} + \frac{8}{\delta\sqrt{2m}} \\
&\leq \epsilon_{\mathcal{D}}(c) + 2\left(\frac{4 + \sqrt{|E| \cdot \log(2emk)}}{\delta\sqrt{2k}} + \frac{8}{\delta\sqrt{2m}}\right)
\end{aligned}$$

Here $\epsilon_S(c, r)$ is simply $\epsilon_S(c) = \frac{1}{k}\sum_{i=1}^{k} \epsilon_{s_i}(c)$. □