



AFRL-RI-RS-TR-2023-137

LEARNING WITH OPTIMAL LABELS

SYSTEMS & TECHNOLOGY RESEARCH

JULY 2023

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2023-137 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

PATRICK FISHER
Work Unit Manager

/ S /

JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE		2. REPORT TYPE		3. DATES COVERED	
JULY 2023		FINAL TECHNICAL REPORT		START DATE AUGUST 2019	END DATE MAY 2023
4. TITLE AND SUBTITLE LEARNING WITH OPTIMAL LABELS					
5a. CONTRACT NUMBER FA8750-19-C-0205		5b. GRANT NUMBER N/A		5c. PROGRAM ELEMENT NUMBER 61101E	
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER R2T1	
6. AUTHOR(S) Dr. Piyush Kumar, Mr. Spencer Douglas					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Systems & Technology Research 600 W Cummings Park Woburn MA 01801				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RISB 525 Brooks Road Rome NY 13441-4505			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI		11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RI-RS-TR-2023-137
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT For Learning with Less Labels Task Area 2, STR team's Learning with Optimal Labels effort developed explicit lower and upper generalization bounds for a general machine learning Model (including complex deep neural networks). It also developed a novel geometric formulation for transfer learning with tight bound on the excess transfer risk, as well as formalism for understanding the performance of different approaches to ML tasks with label constraints.					
15. SUBJECT TERMS Machine Learning, Deep Neural Networks, Statistical Learning Theory, Theoretical Bounds, Transfer Learning, Structured Predictions, Label Constraints					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT		18. NUMBER OF PAGES 48
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			
19a. NAME OF RESPONSIBLE PERSON PATRICK FISHER				19b. PHONE NUMBER (Include area code) N/A	

Table of Contents

List of Figures	ii
1 Summary.....	1
2 Introduction.....	2
3 Methods, Assumptions, and Procedures.....	3
3.1 Generalization Bounds for ML Models	3
3.1.1 Explicit Bounds for Multi-class classification for Simple Models.....	5
3.1.2 Explicit Bounds for Classification for Deep Neural Networks – Fall 2020 Evaluation	6
3.1.3 Explicit Bounds for Object Detection – Fall 2021 Evaluation	12
3.1.4 Explicit Bounds for Video Classification - Final Evaluation	14
3.2 Bounds for Transfer Learning.....	15
3.2.1 Approach based on Gaussian/Rademacher Complexities.....	15
3.2.2 Geometric Formulation for Transfer Learning	20
3.3 Understanding of Structured Prediction and Multi-class classification with Constraints	32
3.3.1 Structured Prediction with Partial Labels and Constraints	33
3.3.2 Generalization Bounds for Constrained Models.....	35
4 Results And Discussion.....	38
5 Conclusions.....	41
6 References.....	42
List of Symbols, Abbreviations and Acronyms.....	43

LIST OF FIGURES

Figure 1: Explicit bounds for the Phishing (Top) and ChemReact (Bottom) datasets as a function of iterations (Left) and Coreset size (Right)	6
Figure 2: Results for Fall 2020 Evaluation - ISI model.....	10
Figure 3: Results for Fall 2020 Evaluation- EI model on Clipart and Sketch datasets	11
Figure 4: Upper and lower bounds for the JPL model on VOC-2009.....	13
Figure 5: Upper and lower bounds for the ISI model on VOC-2009.	13
Figure 6: Generalization Bounds for Video classification for the Berkeley-Kitware Model on the HMDB dataset.	14
Figure 7: Top: Improvement from Transfer Learning vs Learning from Scratch. Bottom: Transfer learning upper bounds.....	19
Figure 8: Excess Transfer Risk Bound for Experiment 1	22
Figure 9: Excess Transfer Risk Bound for Experiment 2	23
Figure 10: Test risks are compared with the Risk bounds for cases - i) with & ii) without transfer learning	25
Figure 11: Empirical distributions of f_{1x} and $f - 1x$ for MNIST	27
Figure 12: Empirical conditional distribution for each class and Q-Q plots for the quantiles	28
Figure 13: Excess transfer risk bounds with the Imagenet shared representation	30
Figure 14: Bound obtained with the sampling approach.	31
Figure 15: Performance on NER task trained with Neural Network Model.....	37
Figure 16: Illustration of the results developed in [18].....	37
Figure 17: Bounds for Image Classification for 4 TA1 models.....	38
Figure 18: Bounds for Object Detection TA1 models.	39
Figure 19: Bounds for Video Classification for Berkeley-Kitware Model.....	39
Figure 20: Upper bound on the excess transfer risk with the geometric approach.....	40

1 SUMMARY

Systems and Technology Research (STR), along with subcontractor MIT, and consultants Prof. Venkat Chandrasekaran and Prof. Alan Willsky, developed *Learning with Optimal Labels (LoL)*, an innovative technology to Learning with Less Labeling (LwLL) Technical Area 2 (TA2). The main objectives of TA2 are:

- Prove tight bounds on amount of labeled data needed to solve machine learning (ML) tasks.
- Leverage prior information about models on related problems.

The key achievements of LwLL are:

- Developed explicit and useful *lower and upper risk bounds* for ML models (including complex DNNs developed by TA1 performers) for many supervised ML tasks:
 - Image Classification, Object Detection, and Video Classification
 - [S. Douglas, P. Kumar, and R. K. Prasanth, “Risk bounds for learning via Hilbert coresets,” JAIR 2022 (under review), arXiv/2103.15569] [1]
- Developed *geometric* formulation of Transfer Learning and construction of novel upper bounds for *excess transfer risk*.
 - [S. Douglas, P. Kumar, V. Chandrasekaran, A. Willsky, “Geometric Bounds for Transfer Learning”, To be submitted][14]
- Developed formalism for understanding precise tradeoff between regularization and inference for *ML tasks with label constraints*.
 - [K. Wang, H. He, T. Nguyen, P. Kumar, D. Roth, “On Regularization and Inference with Label Constraints”, To be published in ICML 2023][18]

2 INTRODUCTION

In order to address the objectives of LwLL TA2, we focused on three research areas:

- Developing upper and lower generalization bounds for the *risk* of a general ML model for a variety of ML tasks like multi-class image and video classification and object detection. This will directly provide bounds on the amount of labeled data needed to solve ML tasks. For this purpose, we developed two approaches – one for upper bounds on the risk and another for lower bounds. First, we built upon the Hilbert coresets formalism developed by our subcontractor Prof. Tamara Broderick [2,3], and developed upper bounds. For the lower bounds, we used the Probably Approximately Correct (PAC)-Bayes formalism as in [7] but generalized it such that it could be used to compute the lower bounds for any ML model.
- With our consultants Prof. Chandrasekaran and Prof. Willsky, we developed a couple of approaches for understanding the theory behind transfer learning, i.e. how leveraging prior information from related problems helps with learning a better more efficient model.
- Finally, based on encouragement from DARPA, we had a fruitful collaboration between Prof. Roth's group at UPenn (who was a TA1 performer) and the STR/MIT group. Together we explored formalisms for understanding structured prediction problems and the problem of multi-class classification with label constraints.

The rest of the report is organized as follows. In Section 3, we will discuss each of the topics mentioned above in detail. In Section 4, we provide results for the various evaluation tests in different phases, followed by discussion. We conclude in Section 5.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Generalization Bounds for ML Models

One of the main achievements of our work during the Learning with Less Labeling program was the development of generalization bounds using Hilbert Coresets. These generalization bounds are one of the main deliverables for TA2 since they provide control on the error on unseen data with prespecified confidence.

These bounds are fully general and computable in the sense that they are applicable to complicated hypothesis classes such as deep neural networks as well as apply to general (even unbounded) loss functions. During the beginning of phase 1, we explicitly computed these bounds for simple models on simple datasets. We describe some of the details of our work below.

We consider the case where the full data set is finite with a total of N samples, i.e. $D := \{X, Y\} = \{(x_n, y_n)\}_{n=1}^N \sim P(x)$. For notational simplicity, we will refer to the probability distribution $p_n = P(x_n)$. The hypothesis is assumed to belong to a hypothesis class H , which is indexed by parameters θ in some parameter space Θ , i.e. $H = \{h_\theta, \text{s.t. } \theta \in \Theta\}$, where $\theta \in \Theta$ is distributed according to some posterior distribution $\nu(\theta) \in \Pi$, where Π is a class of distributions. The parameter space Θ can in principle be infinite, however we will mostly concern ourselves with the finite-dimensional case. The goal is to bound the *expected true risk* which is given by:

$$R(\nu) = E_{\theta \sim \nu} E_{X, Y \sim P} l(h_\theta, x, y) = E_{\theta \sim \nu} \sum_{n=1}^N p_n l_n(h_\theta(x_n), y_n) \quad (1)$$

where $l_n(h_\theta(x_n), y_n)$ is the individual loss for a data-point.

The main idea is to use Hilbert coresets to provide upper bounds on the expected generalization risk, and hence provide an upper bound on sample complexity. Coresets are a sparse weighted subset of data used to summarize a dataset with low reconstruction error -- as measured by a *cost function* $L(\theta) := \sum_n L_n(\theta)$. Hilbert coresets are those that are defined in terms of a Hilbert norm on the space of loss functions. This work [1] builds on the work of Campbell and Broderick [2,3].

We first show that by using the cost function $L(\theta) := E_{X, Y \sim P} l(h_\theta, x, y) = \sum_n p_n l_n(\theta) := \sum_n L_n(\theta)$, $R(\nu)$ can be expressed as the sum of two terms -- i) an empirical term that can be computed from a sparse subset of the full dataset which defines the Hilbert coreset, and ii) an error term introduced by the coreset approximation. We then provide a bound on the error term by generalizing results obtained in [2,3]. The precise result is stated below:

Theorem 3.1: Given a hypothesis space $\mathcal{H}$, a loss function $l : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbf{R}$, for any set Z of size N , by choosing a set S of size $\leq M$ determined by *Algorithm 1*, one has:

$$\forall \nu \text{ on } \mathcal{H}, \forall P \in \mathcal{P}_{CH^{(K)}}, \quad R_{true}(\nu) \leq \|L(\tilde{w})\|_{\nu,2} + \|L(\tilde{w}) - L\|_{\nu,2} \leq \|L(\tilde{w})\|_{\nu,2} + \frac{\hat{\sigma} \hat{\eta} \hat{\eta} \hat{\beta}}{\sqrt{\hat{\eta}^2 \hat{\beta}^{-2(M-1)} + \hat{\eta}^2 (M-1)}}, \quad (12)$$

where $L(\theta) := \sum_{n=1}^N P(x_n) l_n^\theta$, $L(\tilde{w}, \theta) := \sum_{n=1}^N P(x_n) w_n l_n^\theta := \sum_{n=1}^N \tilde{w}_n l_n^\theta$, and the quantities $\hat{\sigma}$, $\hat{\eta}$, $\hat{\eta}$, $\hat{\beta}$, as well as the weights $\tilde{w}$ determined by *Algorithm 1*, depend on the distribution $\nu (\in \mathcal{H})$ and all data ($n = 1, 2, \dots, N$).

Algorithm 1

Given: $\{l_n\}_{n=1}^N$, M , $\langle \cdot, \cdot \rangle$, $\mathcal{P}_{CH}^{(K)}(\hat{p}^{(i)}), i = 1, 2, \dots, K$
 $p_0 \leftarrow \hat{p}^{(i)}$
 $f_0 \leftarrow \arg \max_n \langle \frac{\sigma^{(i)}}{\tilde{\sigma}_n l_n^\theta}, L^{(i)} \rangle$
 $\tilde{w}_0 \leftarrow \frac{\sigma^{(i)}}{\tilde{\sigma}_{f_0}} \mathbf{1}_{f_0}$
for t in $\{1, 2, \dots, M-1\}$ **do**
 $j \leftarrow \arg \max_i \tilde{J}(\tilde{w}_{t-1}, \hat{p}^{(i)})$
 $f_t \leftarrow \arg \max_n \langle \frac{\sigma^{(i)}}{\tilde{\sigma}_n} l_n^\theta, L^{(i)} - l(\tilde{w}_{t-1}) \rangle$
 $\gamma_t \leftarrow \frac{\langle \frac{\sigma^{(i)}}{\tilde{\sigma}_{f_t}} l_{f_t}^\theta - l(\tilde{w}_{t-1}), L^{(i)} - l(\tilde{w}_{t-1}) \rangle}{\left\| \frac{\sigma^{(i)}}{\tilde{\sigma}_{f_t}} l_{f_t}^\theta - l(\tilde{w}_{t-1}) \right\|^2}$
 $\tilde{w}_t \leftarrow (1 - \gamma_t) \tilde{w}_{t-1} + \gamma_t \frac{\sigma^{(i)}}{\tilde{\sigma}_{f_t}} l_{f_t}^\theta$
end for
return $\tilde{w}_{M-1}$

Some of the main features of this theorem are as follows:

- It generalizes result in [2,3] for all data distributions P that lie in a convex polytope determined by K points: $\mathcal{P}_{CH}^{(K)}$, by solving a robust (*min-max*) optimization problem
- It provides efficient summarization of data for supervised learning when all labeled data is provided:
 - Coreset size (# of nonzero weights $\tilde{w}$) is much smaller than M
 - The second term converges exponentially (β^M) for moderate M ($1/\sqrt{M}$ for small M)
 - It is true for any real-valued (possibly unbounded) loss function
 - Parameters in the ‘complexity’ term $\{\hat{\sigma}, \hat{\eta}, \hat{\eta}, \beta\}$ can be computed from individual datapoint losses & given $\langle \cdot, \cdot \rangle$

The above theorem is true for an exact norm, which is hard to compute in general. However, for loss functions l such that $l_n l_m$ is sub-Gaussian with parameter ξ^2 , we have with probability $\geq$

$1 - \delta$ for J samples, we show that the exact norm is bounded by the approximate norm (computed by Monte-Carlo methods) and an additional term:

$$\begin{aligned}
 \|L(\tilde{w}) - L\|_{v,2}^2 &\leq \|\hat{L}(\tilde{w}) - \hat{L}\|_{v,2}^2 + \sup_{P \in \mathcal{P}_{CH}^{(K)}} \|\tilde{w} - p\|_1^2 \sqrt{\frac{2\xi^2}{N} \log \frac{2N^2}{J}} \\
 &= \|\hat{L}(\tilde{w}) - \hat{L}\|_{v,2}^2 + \|\tilde{w} - p^{(I)}\|_1^2 \sqrt{\frac{2\xi^2}{N} \log \frac{2N^2}{J}},
 \end{aligned} \tag{31}$$

We obtain this result by extending the result in [1, 2] for all data distributions P that lie in the convex polytope $\mathcal{P}_{CH}^{(K)}$. This result allows us to explicitly compute bounds for various datasets and models.

3.1.1 Explicit Bounds for Multi-class classification for Simple Models

Using the results above, we constructed explicit bounds. As mentioned earlier, we first chose simple datasets and models for our preliminary analysis. We choose a logistic regression model and negative log-likelihood as the loss function. We choose the ‘generalized posterior’ distribution $v(\theta)$ as the Laplace approximation to the true posterior distribution, the latter of which is found by Monte-Carlo methods. A uniform distribution on the data is assumed. We choose two simple binary classification datasets – i) *Phishing* [4], ii) *ChemReact* [5]. The *Phishing* data set consisted of $N = 11,055$ data points each with $D = 68$ features. In this data set, each covariate corresponds to the features of a website, and the goal is to predict whether or not a website is a phishing site. The *ChemReact* data set consisted of $N = 26,733$ data points each with $D = 10$ features. In this data set, each covariate represents the features of a chemical experiment, and the label represents whether a chemical was reactive or not.

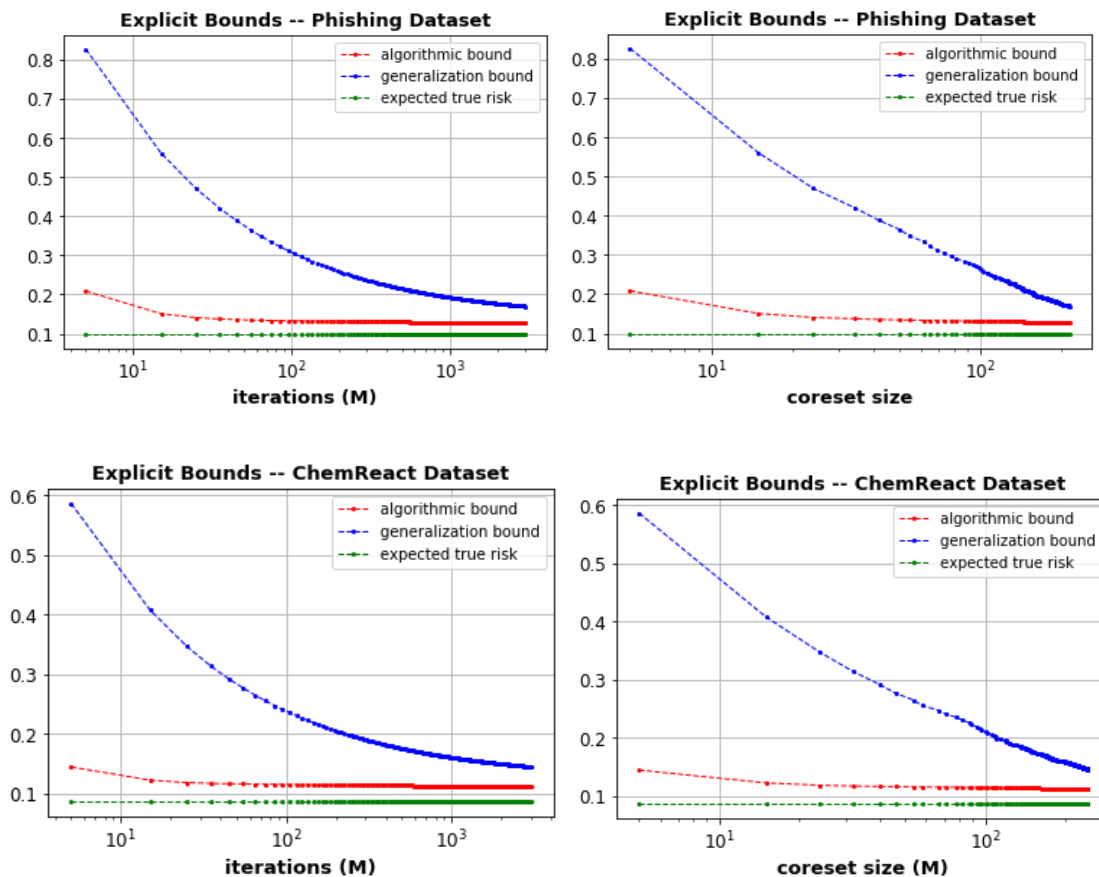


Figure 1: Explicit bounds for the Phishing (Top) and ChemReact (Bottom) datasets as a function of iterations (Left) and Coreset size (Right)

From the figure, we confirm that the bounds indeed decay exponentially with the number of iterations and that the coreset size is typically much smaller than the number of iterations. Thus, these provide very efficient bounds.

3.1.2 Explicit Bounds for Classification for Deep Neural Networks – Fall 2020 Evaluation

For the Fall 2020 TA2 evaluation task, we were provided with the following datasets from DARPA:

- Task ID: domain-net-clipart $\rightarrow$ domain-net-sketch
- Task ID: voc2009 $\rightarrow$ pool-car-detection

The first task above is a *multi-class classification* task while the second one is an *object detection* task. For this evaluation, we focused on the multi-class classification task.

Since the goal of TA2 is to develop theoretical bounds for hypotheses (ML models) developed by TA1 performers, we approached a number of TA1 performers to get information about their models. In particular, the performers we contacted are:

- Prof. A. Galstyan’s group at ISI
- Dr Y. Eisenberg’s group (Embedded Intelligence)
- IBM
- UC Berkeley, Kitware

Encouraging responses were received from Prof Galstyan’s and Dr Eisenberg’s groups, and we had interesting discussions with both. We received the following trained models:

- *Information Sciences Institute* (ISI) provided us with a trained model for the CIFAR-100 dataset. In addition to providing the model architecture, they provided the learned parameters of the trained model (at various checkpoints), as well as the precise training data (at various checkpoints) on which their model was trained. They also provided a test set on which their trained model was evaluated. Very recently they also provided models for the domain-net-clipart/domain-net-sketch datasets, but we have not got the chance to compute bounds for these.
- *Embedded Intelligence* (EI) provided us with a trained model for the domain-net-clipart & domain-net-sketch datasets. They also provided the model architecture, learned parameters of the trained model (at various checkpoints), and a test set on which their models were evaluated. However, they did *not* provide us the precise training set on which their models were trained, only the number of training samples at each checkpoint was provided. This leads to some uncertainty in the calculation of bounds for the EI models as will be explained later. Note that this is not a theoretical issue, just a practical consequence of not having access to the precise training sets.

We presented results for generalization bounds for the *Embedded Intelligence* (EI) trained model on the domain-net-clipart and domain-net-sketch datasets, and for the *Information Sciences Institute* (ISI) trained model on the CIFAR-100 dataset in our Evaluation report submitted to DARPA in September 2020.

3.1.2.1 Setup for Generalization Bounds

The setup for computing generalization bounds is that of statistical learning theory. In order to provide theoretical upper and lower bounds on a learning problem using the methods of *statistical learning theory*, one requires the following ingredients:

- A training dataset $Z^m = \{X_i, Y_i\} \in \mathcal{X} \times \mathcal{Y}, i = 1, 2, \dots, m$
 - **Fall 2020 Evaluation:** i) domain-net-clipart, ii) domain-net-sketch; iii) cifar-100 datasets.
- Hypothesis class $\mathcal{H} = \{h(\cdot, \theta): \mathcal{X} \rightarrow \mathcal{Y} \mid \theta \in \Theta\}$
 - **Fall 2020 Evaluation:** Architecture and parameters of the deep neural network (DNN) trained by the following TA1 models.
 - **Example 1:** Information Sciences Institute (ISI) model on CIFAR-100 dataset [Prof. Galstyan’s group]
 - **Example 2:** Embedded Intelligence (EI) models on domain net clipart–domain net sketch datasets [Y. Eisenberg’s group]
- Input data distribution $D \in \mathcal{D}, D$ is unknown in general.
- Loss function: $l = \mathcal{Y} \times \mathcal{Y} \rightarrow \mathcal{R}$
 - 0-1 loss function, directly related to accuracy score on test set

With the above ingredients in place, the goal of generalization bounds is to provide guarantee on performance of a learning algorithm on unseen data with prespecified confidence. More precisely, the goal is to bound the *true risk*:

$$R(h_\theta) = E_{Z \sim D} l(h_\theta(x), y) \quad (2)$$

The metric to measure the performance of TA1 performers for multi-class classification is the *accuracy score*, which can be easily related to the risk on the test set, or the *test risk* via:

$$\text{test risk} = 1 - \text{accuracy score} \quad (3)$$

which is a stand-in for the *true risk*.

In our work below, we focus on a *PAC-Bayes* based approach, where one starts with a data-independent parameter distribution over parameters θ on hypothesis space ($\mathbf{P}$), and the learning algorithm is used to construct data-dependent and algorithm-dependent parameter distribution on hypothesis space ($\mathbf{Q}$). Then, one studies the *expected true risk* by averaging over $\mathbf{Q}$:

$$R(Q) = E_{h \sim Q} R(h) \quad (4)$$

Contrary to VC dimension and Rademacher complexity bounds, these are **non-uniform** bounds in $\mathcal{H}$, i.e. are valid in neighborhood of parameters learned by (TA1) learning algorithm. The non-uniformity of bounds in hypothesis space allows bounds to be much tighter and useful. We develop two kinds of explicit generalization bounds in the following, both of which are *non-uniform* and bound the expected true risk.

3.1.2.2 Generalization Upper Bounds via Hilbert Coresets

We first discuss the generalization upper bound developed by us using the method of Hilbert coresets. The formalism behind this approach was described in detail in section 3.1. The bound calculation workflow consists of the following steps:

- Choose an appropriate space for the data distributions D . For simplicity, we choose a trivial convex hull consisting of a single point. This amounts to choosing a uniform distribution on the data.
- Choose appropriate parameter distributions $\mathbf{P}$ and $\mathbf{Q}$ (this will be discussed in section 3.1.2.4)
- Given $\mathbf{Q}$, compute an approximation to the Hilbert space inner product $\langle \cdot, \cdot \rangle$ using the method of random projection features [6]
- Then use *Algorithm 1* above to compute the coreset weights $\tilde{w}$ and the constants in the RHS of theorem above.
- Compute bounds based on the RHS of theorem above.

3.1.2.3 Lower and Upper Bounds for Supervised Learning à la Begin et al [7]

In addition to the upper bounds computed from Hilbert Coresets, we also compute lower and upper bounds via the standard PAC-Bayes approach. This is done for two reasons. First, these provide an alternative way to compute upper bounds which can be compared to the Coreset upper bounds. In addition, by using a modern version of the PAC-Bayes approach by Begin *et al* [7], we can also compute lower bounds on $R(Q)$.

In particular, the bound derived by Begin *et al* puts an upper bound on a convex distance function Δ between the expected true risk and the expected empirical risk:

Theorem 5.1 [1]: For a training set with data distribution $D \in \mathcal{D}$, for any ‘prior’ distribution P on $\mathcal{H}$, for any $\delta \in (0, 1]$, for any convex function $\Delta : [0, 1] \times [0, 1] \rightarrow \mathbf{R}$, for any $m' > 0$, with probability at least $1 - \delta$ over the choice S with m input-output pairs, one has for all posterior distributions Q on $\mathcal{H}$:

$$\Delta(R(Q), \hat{R}_S(Q)) \leq \frac{\left[KL(Q||P) + \log\left(\frac{I_{\Delta}^K(m, m')}{\delta}\right) \right]}{m'}, \quad (9)$$

The upper bound on convex Δ provides both lower and upper bound on expected true risk $R(Q)$ (with high probability):

$$\hat{R}_S(Q) - \Delta^{-1} \left(\frac{\left[KL(Q||P) + \log\left(\frac{I_{\Delta}^K(m, m')}{\delta}\right) \right]}{m'} \right) \leq R(Q) \leq \hat{R}_S(Q) + \Delta^{-1} \left(\frac{\left[KL(Q||P) + \log\left(\frac{I_{\Delta}^K(m, m')}{\delta}\right) \right]}{m'} \right) \quad (5)$$

Again, the bound calculation workflow consists of the following steps:

- Choose appropriate parameter distributions (data-independent) P and (data & TA1 algorithm dependent) Q [see section 3.1.2.4]
- For a given training set size m , optimize S over $\{\Delta, m'\}$
- Estimate expected empirical risk $\hat{R}_S(Q)$ using Monte-Carlo approximation.
- Compute lower and upper bounds using theorem above.

3.1.2.4 Results for Fall 2020 TA2 Evaluation

Before presenting our results below, we discuss our choice of the parameter distributions P and Q . As discussed above, the bounds are valid for all parameter distributions P & Q , but an appropriate choice of these bounds can lead to tight bounds. In particular, our choice is the following:

- $\mathbf{P}$ is chosen as multivariate normal distribution $N(0, \sigma^2 \cdot I)$ – data independent
- $\mathbf{Q}$ is chosen as multivariate normal distribution $N(\theta^+, \Sigma_{\theta^+})$, determined by TA1 learning algorithm (that learns θ^+). $\Sigma_{\theta^+}^{-1} = \max [\text{Hessian of loss function around } \theta^+, \sigma^{-1}]$ – training data & learning algorithm dependent. This choice is inspired from the work of Banerjee *et al* [8]
- Optimize over σ

3.1.2.4.1 Results for the ISI Model on the CIFAR-100 Dataset

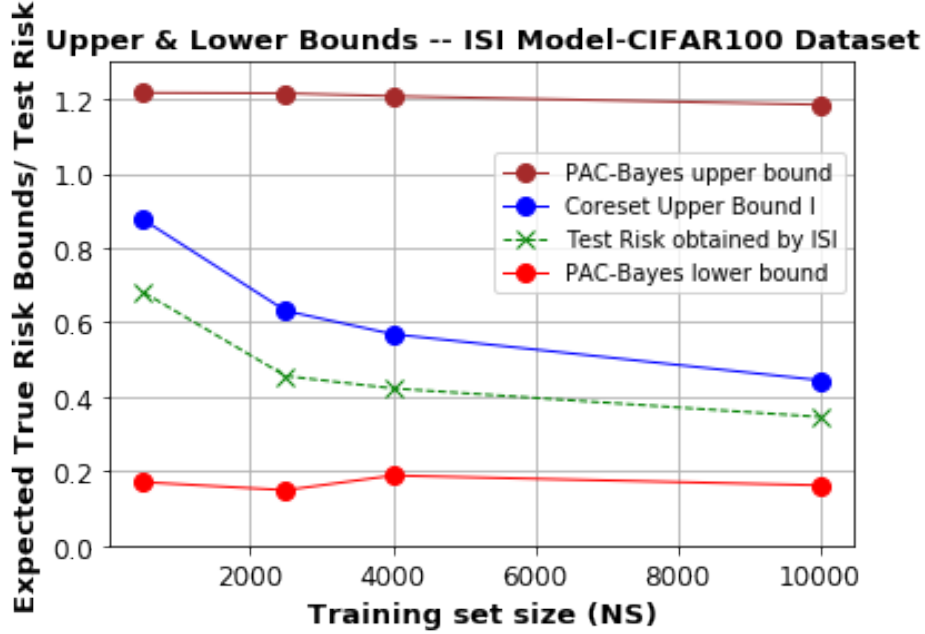


Figure 2: Results for Fall 2020 Evaluation - ISI model

The results for both kinds of bounds for the ISI model on the CIFAR-100 dataset are presented in the *brown*, *blue* and *red* curves in the figure above. In addition, the *green* curve shows the empirical risk on the test set (or the test risk) which can be readily obtained from the accuracy score of the ISI model on the test set. One can see that the green curve is nicely bounded from above by the Coreset upper bound (blue) and from below by the PAC-Bayes lower bound (red). The coreset upper bound is much tighter than the PAC-Bayes upper bound.

Some other points are in order. For the Coreset upper bound, we choose $\sigma = 10^{-4}$, for which it can be shown that the posterior distribution $\mathbf{Q}$ is sharply peaked around the learned parameters θ^+ . This means that the Coreset upper bound is an effectively deterministic bound. On the other hand, the PAC-Bayes bounds in the figure above use choose $\sigma = 0.5$, for which $\mathbf{Q}$ is not as sharply peaked. Hence the PAC-Bayes bounds are *stochastic* in nature, i.e. they are valid for parameters averaged over *neighborhood* of learned parameters θ^+ . If we make σ much smaller than 0.5, the PAC-Bayes bounds become vacuous, for example they give negative lower bounds. Although the $\sigma = 0.5$ bounds are stochastic, they still provide useful lower bounds as can be seen from the figure above.

The results for the EI model on the Domain-net-clipart and Domain-net-sketch datasets are shown in the figures below. In this case, there is an extra complication that arises from the fact that the precise training set checkpoints that were used to train the EI model on the above datasets were *not* available to us. Therefore, we had to create training checkpoints of the appropriate size by randomly sampling from the full dataset. Because of this, there is a source of uncertainty in posterior computation and hence in the computation of the bounds as well. Even with this uncertainty, our Coreset upper bound results seem very close to the actual test risk obtained by the EI models. As before, the Coreset upper bounds are much tighter than the PAC-Bayes upper bounds. Also, as before, the Coreset upper bounds are effectively deterministic (with $\sigma = 10^{-6}$), while the PAC-Bayes upper and lower bounds are stochastic (with $\sigma = 0.03$). Nevertheless, the PAC-Bayes bounds in the figures below still provide useful bounds.

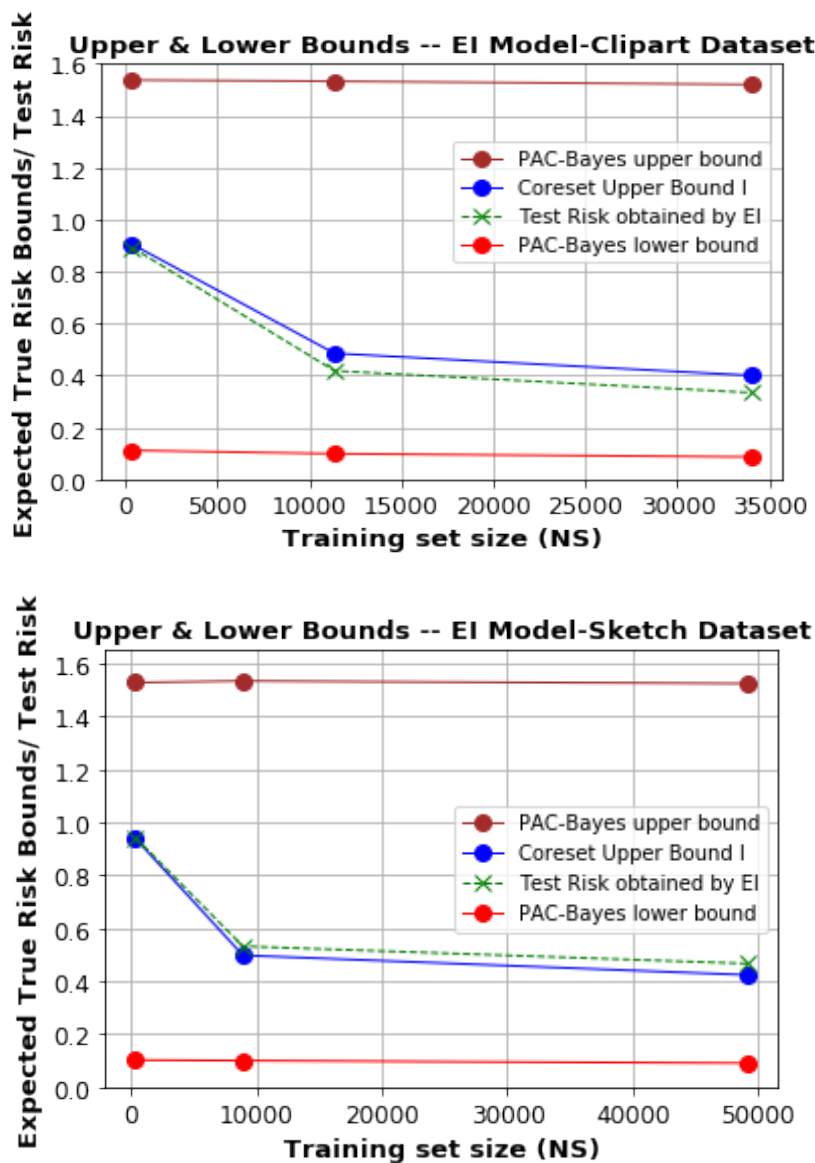


Figure 3: Results for Fall 2020 Evaluation- EI model on Clipart and Sketch datasets

3.1.3 Explicit Bounds for Object Detection – Fall 2021 Evaluation

In the Fall 2020 evaluation, we successfully developed bounds for image classification tasks for the domain-net-clipart and domain-net-sketch DARPA development datasets. For the Fall 2021 evaluation, we focused on developing bounds for object detection tasks. We use the same formalism as developed for the multi-class classification task.

While the formalism is the same, the details are somewhat different. One of the key differences between the classification and object detection cases is the loss function. A loss function l measures the ‘goodness’ of a hypothesis $h \in H$ by comparing the predicted output with the true output, $l: Y \times Y \rightarrow R$. For object detection, we are mostly concerned with the mAP loss function $l_{mAP} = 1 - mAP$, where mAP stands for the mean average precision score.

For Fall Evaluation 2021, we developed explicit lower and upper bounds for object detection for two models described below:

- Jet Propulsion Labs (JPL) provided us with a trained baseline model for the Vis-drone dataset.
- Information Sciences Institute (ISI) provided us with a trained model for the VOC-2009 datasets.

In addition to providing the model architecture, the above teams provided the learned parameters of the trained model (at various checkpoints), as well as the precise training data (at various checkpoints) on which their model was trained. They also provided a test set on which their trained model was evaluated. We will present results for lower and upper bounds for the JPL trained baseline model on the *Vis-drone* dataset, and for the Information Sciences Institute (ISI) trained model on the VOC-2009 dataset.

3.1.3.1 Generalization Bounds for Object Detection

Object detection belongs to a class of structured learning problems that are more challenging than image classification problems. There are many reasons for this:

- Task loss for object detection, i.e. the mAP loss defined earlier, is more complicated than simple 0/1 loss for classification. In particular, the mAP loss is a real number between 0 and 1 and is computed from a complex calculation.
- Computing predictions requires inference over potential structures and is part of the learning process, making it computationally hard in general. This is in contrast to image classification where computing predictions requires a simple argmax over a set of categories and is trivial. Because of the complexity above, most work on training object detection models involves surrogate regression and classification tasks followed by post-processing steps that are mostly heuristic. Furthermore, unlike image classification, where the loss used during training models (e.g. cross-entropy loss) is a good proxy for the 0-1 loss (with theoretical guarantees), these surrogate training losses do not have any guarantees.

Recently, there has been some work on object detection that tries to follow a more principled set prediction approach, see [9,10]. However, both the JPL and the ISI models provided to us used different versions of surrogate training losses during training. In the future, it will be useful to compute bounds for the more principled object detection models.

3.1.3.2 Results for Object Detection Bounds

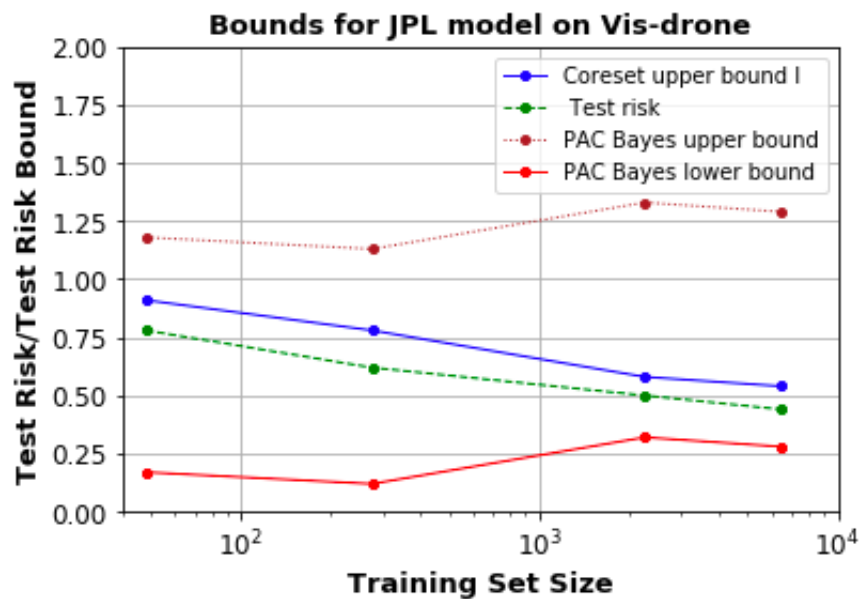


Figure 4: Upper and lower bounds for the JPL model on VOC-2009.

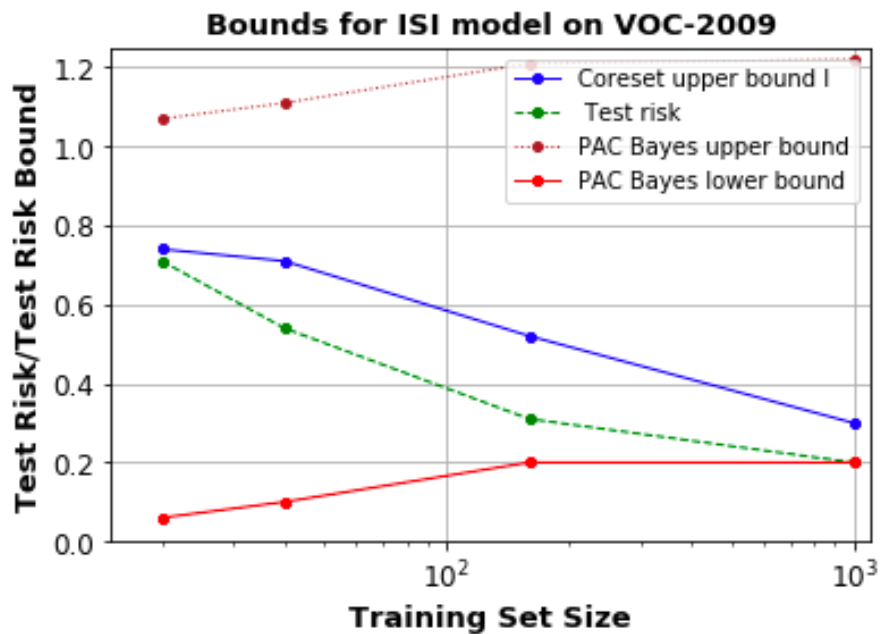


Figure 5: Upper and lower bounds for the ISI model on VOC-2009.

Various observations can be made about the results above:

- It turns out that Coreset upper bounds are much better than PAC-Bayes upper bounds – we only show the former for the upper bounds.
- A prior std. dev (σ) $\ll 1$ was used for upper bounds, so upper bounds are effectively deterministic.
- For lower bounds, small (σ) $\ll 1$ does not lead to useful bounds. Therefore, we used $\sigma = 6$ to obtain useful results, so lower bounds are stochastic both in principle and in practice
- The lower bounds (PAC-Bayes) for both the JPL & ISI models are non-monotonic with training size. This behavior is very different from that in image classification case. The reason is that the bound depends on expected empirical risk (on training set) $\widehat{R}_S(Q)$, which is non-monotonic with training size for such models. We expect that a training model that uses a theoretically well-motivated loss for object detection (in contrast to those used for the ISI and JPL models) should have a monotonic behavior. This will be interesting to confirm in future experiments.

3.1.4 Explicit Bounds for Video Classification - Final Evaluation

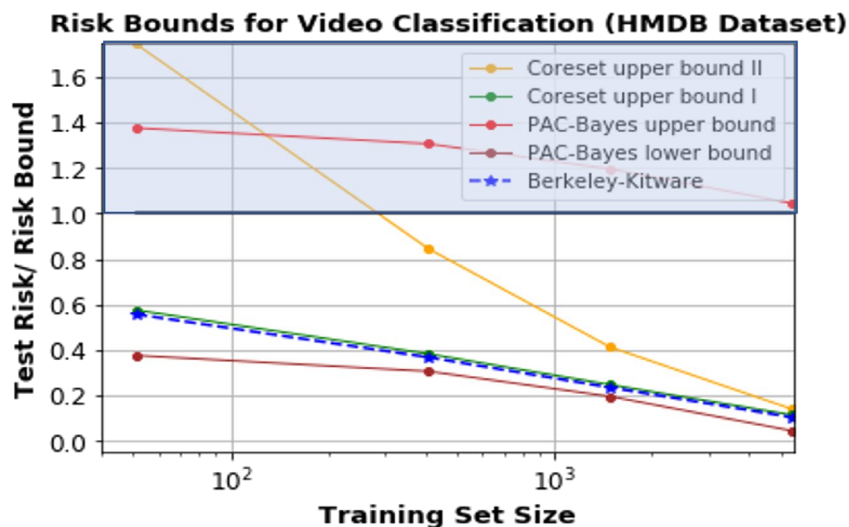


Figure 6: Generalization Bounds for Video classification for the Berkeley-Kitware Model on the HMDB dataset.

For the final evaluation, we developed bounds for another class of ML tasks- video classification, using the same formalism. As for the previous evaluations, we requested models from TA1 performers. We received one model – from Berkeley-Kitware, on the Human Motion Database (HMDB) dataset, which has video clips from various movies with 51 labeled actions, e.g., walking, running, chewing, etc.

The result for the generalization bounds for video classification is shown in **Figure 6**.

3.2 Bounds for Transfer Learning

In the previous section, we described the research conducted by the LoL Team on Generalization bounds for supervised ML tasks for case when the models were trained from scratch with some amount of training data. However, most of the modern applications of ML rely on transferring knowledge from related tasks to the task at hand. This approach has empirically shown great results and promise. Therefore, as part of our research for the LwLL program, we also developed new approaches for transfer learning.

3.2.1 Approach based on Gaussian/Rademacher Complexities

Our consultant - Prof. Chandrasekharan, developed a new formulation of transfer learning, using the *inverse problem* approach that was inspired by the work of [11, 12]. The idea is to find a common “environment” that codifies all the features of interest in a collection of related learning problems. Since each learning task depends on different subsets of these features, the goal of transfer learning is to identify this common environment among a collection of related tasks.

Building upon the above idea, the following setup for transfer learning is considered. We are given labeled data $\{x_i^{(t)}, y_i^{(t)}\}, i = 1, 2, \dots, m$ for a learning task (denoted the *target* problem, hence the superscript t) and we wish to learn a suitable mapping $f^{(t)}$ so that $f^{\{(t)\}}(x_i^{(t)}) \approx y_i^{\{(t)\}}, i = 1, \dots, m$. As these labeled data are expensive to obtain, the number of observations m is relatively small. To alleviate this difficulty, we are also given labelled data $\{x_i^{(s)}, y_i^{(s)}\} \subset R^d \times R, s = 1, \dots, S$, corresponding to a number of other related tasks (these are called *source* tasks). Labeled data for these source tasks are inexpensive to obtain so n is large for each $s \in S$. (The number of samples n could vary across the source tasks, but we will focus on this simplified situation to minimize notational clutter.) The goal is to leverage commonalities in the labelled data across the source and target tasks to learn a better mapping $f^{(t)}$ for the target task than would be possible without access to the source tasks.

Our approach leverages the results of Jordan *et al* [13]. [13] tries to bound the excess transfer risk for the classifier function ($\hat{f}$) learnt on the target task for the best representation ($\hat{h}$) learnt from the S source tasks. The main computational techniques employed are the following:

- Estimation of Gaussian complexities of the representation class (H) and that of the function class (F)
- Identifying conditions under which the source tasks are “diverse” enough so that the representation learned from the source tasks is also good for the target task, when composed with the task function.

In particular, [13] bounds the excess transfer risk as follows when assumptions (1) and (2) stated below are satisfied.

Assumption 1: The loss function $l(\cdot)$ is *bounded* and *Lipschitz*. Any function f in the function class F is $L(F)$ -*Lipschitz*, and the composition $(f \circ h)$ is *bounded* as well.

Assumption 2: The true representation h^* is contained in H . Additionally, the true task specific functions f_j^* are contained in F for both the S training tasks and the target test task.

Theorem 1.1 When assumptions about regularity of the loss function (Assumption 1), and realizability of the learning setup (Assumption 2) hold, and if the training tasks are (ν, ϵ) diverse, then with probability $1 - 2\delta$, the transfer risk is bounded by:

$$R_{test}(\hat{f}_0, \hat{h}) - R_{test}(f_0^*, h^*) \leq 4096L \log(nS) \left(\frac{L(\mathcal{F})\mathfrak{G}_{nS}(\mathcal{H}) + \mathfrak{G}_n(\mathcal{F})}{\nu} \right) + 16L\mathfrak{G}_m(\mathcal{F}) + \frac{4096LD_X}{\nu(nS)^2} + 8\beta \left(\frac{1}{\nu} \sqrt{\frac{\log(\frac{2}{\delta})}{nS}} + \sqrt{\frac{\log(\frac{2}{\delta})}{m}} \right) + \epsilon \quad (6)$$

In the above theorem, $\hat{f}_0$ is the empirical risk minimizer classifier for the task function class given the learned representation $\hat{h}$ over all the source tasks.

In our work so far, we have been able to apply the general formalism of Tripuraneni *et al* [13] to a particular case, where $\mathcal{H}$ = PCA representation class and $\mathcal{F}$ = logistic regression class. The relevant quantities that need to be computed are shown below:

For PCA, the representation class is

$$\mathcal{H} = \{h(x) = Bx | B \in \mathbb{R}^{K \times d}, B \text{ has orthonormal rows} \} \quad (19)$$

and task class

$$\mathcal{F} = \{f(z) = a^T z | a \in \mathbb{R}^K, \|a\| \leq c\} \quad (20)$$

For PCA, we typically have K much smaller than d . We define the data matrix X as the d by N matrix of data points x_i . Note this means the matrix $\tilde{q} = BX$ is the K by N matrix from the definition of Gaussian complexity, where $J = K$. During the training phase, we have S source tasks each of which have a sample of size n (for simplicity), so $N = nS$ for representation (feature) learning and $N = n$ for source task learning. During the test phase, we have the target task with m examples, so $N = m$ for target task learning.

We need to compute the Gaussian complexity for the representation class, given by:

$$\mathfrak{G}_{nS}(\mathcal{H}) = \mathbb{E}_X \mathbb{E}_g \sup_{B \in \mathcal{H}} \frac{1}{N} \text{Tr}(g^T BX); \quad N = nS \quad (21)$$

For a task class (any one of the S source tasks with sample size n or the target task with sample size m) $\mathfrak{G}_{n|m}(\mathcal{F})$, we have $\tilde{q} = a^T BX$ and $J = 1$. This gives:

$$\mathfrak{G}_{n|m}(\mathcal{F}) = \mathbb{E}_X \left[\max_{\{X\}_{n|m} \in \mathcal{X}} \max_{B \in \mathcal{H}} \mathbb{E}_g \sup_{a \in \mathcal{F}} \frac{1}{N} \text{Tr}(g^T a^T BX) \right] = \mathbb{E}_X \left[\max_{\{X\}_{n|m} \in \mathcal{X}} \max_{B \in \mathcal{H}} \mathbb{E}_g \sup_{a \in \mathcal{F}} \frac{1}{N} (gX^T B^T a) \right]; \quad N = n \text{ or } m \quad (22)$$

We have been able to explicitly estimate the gaussian complexities for the above representation class and the task class, as well as the other quantities. Substituting the results for these and the other quantities in Theorem 1.1, we get the following result:

$$R_{test}(\hat{f}_0, \hat{h}) - R_{test}(f_0^*, h^*) \leq 4096L \log(nS) \left(\frac{\frac{cK}{\sqrt{nS}} \sqrt{d\|\Sigma\|} + 2d\delta \sqrt{\frac{C_K}{nS} \log \frac{1}{\delta}} + \mathbb{E}_X[\max_{X_n \in \mathcal{X}} \frac{c}{\sqrt{n}} \sqrt{\sum_{i=1}^K \lambda_i(\Sigma_X^S)}]}{\tilde{\nu}} \right) + 16L \mathbb{E}_X \left[\max_{X_m \in \mathcal{X}} \frac{c}{\sqrt{m}} \sqrt{\sum_{i=1}^K \lambda_i(\Sigma_X^T)} \right] + \frac{4096LcD}{\tilde{\nu}(nS)^2} + 8\beta \left(\frac{1}{\tilde{\nu}} \sqrt{\frac{\log(\frac{2}{\delta})}{nS}} + \sqrt{\frac{\log(\frac{2}{\delta})}{m}} \right)$$

The above result is valid as long as the source tasks are (ν, ϵ) diverse. For the PCA-logistic regression case, task diversity is guaranteed as long as certain conditions are satisfied. We are able to show the following result:

Assumption 3: If $\lambda_K(\frac{A^T A}{S}) = \bar{\nu} > 0$ where λ_K is the K^{th} eigenvector and A a matrix of vectors from $\mathcal{F}$, then it is shown that the tasks are $(\Omega(\bar{\nu}), 0)$ diverse in the sense of Definition 3 in [1].

We then moved on to computing bounds for two other interesting representation and function classes:

- Sparse-coding representation/function class, &
- Neural network representation class with logistic function class

We computed the Gaussian complexities for the *sparse coding* and *neural network representation class*, and also computed the *task diversity* parameter for a set of source and target tasks created from the MNIST dataset, and computed transfer risk bounds based on the above approach

While computing explicit bounds for the MNIST dataset with the above Gaussian complexities, we found that the *task diversity* criterion used in the transfer learning formalism of Tripuraneni *et al* [13] is not a suitable criterion for transfer learning. Therefore, we also studied the transfer learning formalism of Maurer *et al* [12] to develop explicit transfer risk bounds for the MNIST dataset. We obtained non-vacuous bounds for this case for the PCA representation class and the logistic classifier task class. We identified *pros* and *cons* of both formalisms and laid out a path forward for our approach that combines the best of both.

As discussed earlier, the results for the Gaussian complexities for the sparse coding and neural network representation class are shown below. The relevant quantities that need to be computed for the *sparse coding* case are:

In this case, we have the representation class

$$\mathcal{H} = \{h(x) = Bx | B \in \mathbb{R}^{(pxp) \times d}, B = [B_1 | B_2 | \dots | B_d], B_i \in \mathbb{R}^{pxp}, \sum_{i=1}^d B_i^T B_i = \sum_{i=1}^d B_i B_i^T = I\}, \quad (21)$$

and the task class

$$\mathcal{F} = \{f(Z) = \text{Tr}(AZ) | A \in \mathbb{R}^{pxp}, \|A\|_F \leq c, \text{rank}(A) \leq r\} \quad (22)$$

So we need the following complexities:

$$\begin{aligned} \mathbb{G}_{nS}(\mathcal{H}) &= \mathbb{E}_X \mathbb{E}_g \sup_{B \in \mathcal{H}} \frac{1}{N} \text{Tr}(g^T B X); \quad N = nS \\ \bar{\mathbb{G}}_{nm}(\mathcal{F}) &= \mathbb{E}_X \left[\max_{\{X\}_{n|m} \in \mathcal{X}} \max_{B \in \mathcal{H}} \mathbb{E}_g \sup_{a \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N g_i \text{Tr}(ABx_i) \right]; \quad N = n \text{ or } m \end{aligned} \quad (23)$$

In the case of a low-rank matrix, we generally have $p^2 \geq d$ and $r < d$. Similar to the sparse coding case, this means we have a large number of components to choose from, but there are only a small number of components which are actually relevant to the task. For the task class, we define

$$\begin{aligned} L(\mathcal{F}) &= \sup_A \|A\|_F = c \\ D_X &= \sup_{A, B, x} \text{tr}(ABx) \end{aligned} \quad (24)$$

Using techniques similar to that for the PCA case, we arrive at the following bound for the representation complexity:

$$\mathfrak{G}_{nS}(\mathcal{H}) \leq \frac{K}{\sqrt{nS}} \sqrt{d\|\Sigma\| + 2d\delta \sqrt{\frac{C_K}{nS} \log \frac{1}{\delta}}}$$

Similarly, the relevant quantities that need to be computed for the neural network case are:

For computing bounds on neural networks, we will consider C -layer neural networks. For each layer, we have a matrix of parameters W_c , with appropriate dimension. We also have an activation function, $\sigma(x)$, for each layer. The model has a single output, and can be written as

$$y\hat{y} = a^T W_C \sigma(W_{C-1} \sigma(\dots (W_1 x) \dots)) \quad (25)$$

For the matrices W_c , we define the norm

$$\|W_c\|_{1,\infty} = \max_j \sum_i |w_{j,i}| \quad (26)$$

We also have the induced operator norm

$$\|W_c\|_{\infty \rightarrow 2} = \max_x \frac{\|W_c x\|_2}{\|x\|_\infty} \quad (27)$$

For a neural network, we will define the representation set to be

$$\mathcal{H} = \{h(x) = W_C \sigma(W_{C-1} \sigma(\dots (W_1 x) \dots)) \mid \|W_c\|_{1,\infty} \leq M(c), 1 \leq c < C, \max(\|W_C\|_{1,\infty}, \|W_C\|_{\infty \rightarrow 2}) \leq M(C)\} \quad (28)$$

The task class is

$$\mathcal{F} = \{f(z) = a^T z \mid \|a\|_1 \leq c_1 M(C)^2\} \quad (29)$$

In order to upper bound the Gaussian complexity for this case, we leverage the results of Golowich *et al* [4]. Then we can compute the following results:

$$\hat{\mathfrak{G}}_X(\mathcal{H}) \leq K \log nS \frac{2MD \sqrt{C+1+\log d}}{\sqrt{nS}} \quad \hat{\mathfrak{G}}_X(\mathcal{F}) \leq \frac{c_2 M(C)^2}{\sqrt{m}} \max_i \|h(x_i)\| \quad (6,7)$$

As discussed earlier, we then moved on to computing explicit transfer risk bounds for the MNIST dataset. To set up the experiment, we divided the MNIST dataset into a set of twelve source tasks and one target task. Each task was a binary classification task of two digits. We developed an algorithm to learn the shared representation, followed by an algorithm to learn the target task function. The performance of this algorithm was then compared with the transfer risk bounds which we computed. In order to compute the bounds, we first started with the PCA case for simplicity. After computing the relevant Gaussian representation and task complexities, we computed the *task diversity* parameter v . However, applying the precise criterion for task diversity for the PCA case as in Tripuraneni *et al* [13], we found that the task diversity was much smaller than 1. This made the transfer bounds very loose and not useful in practice. On the contrary, the transfer learning algorithm described above was able to show very good performance on the transfer task. This shows that the task diversity criterion in Tripuraneni *et al* [13] does not capture the notion of *relevance* of the source tasks to transfer task well. This is one of the areas we have identified to explore further.

Maurer et al [12] follow the same general philosophy of Baxter [11] (as does [13]) that the source and target tasks lie within a common ‘environment’. However, unlike [13], Maurer et al [12] also assume that the source and target tasks are sampled *i.i.d* from the environment. This allows them to derive bounds for transfer risk without any notion of task diversity. Adapting their results to the case at hand, we get the following result for the transfer risk:

$$R_{\mu}(\hat{h}) \leq \frac{1}{m} \sum_{i=1}^m \ell(\hat{f}_{\mathbf{Z}}(h(x_i^i)), y_i^i) + \frac{\sqrt{2\pi}}{m(S+1)} \sum_{j=1}^{(S+1)} \sup_{h \in \mathcal{H}} \hat{\mathcal{G}}(\mathcal{F}(\hat{h}(\mathbf{x}_t), y_t)) + \sqrt{\frac{\log(2/\delta)}{2(S+1)}} + \sqrt{\frac{\log(2/\delta)}{2m}}$$

We explicitly computed the quantities in the RHS of the inequality above for the PCA representation class and the logistic classifier task class to upper bound the transfer risk for the MNIST dataset. This was then compared with the transfer learning algorithm described in the previous section. The results are shown in Figure 3:



Figure 7: Top: Improvement from Transfer Learning vs Learning from Scratch. Bottom: Transfer learning upper bounds

As can be seen from the bound above, there is a $\sqrt{\frac{1}{T}}$ suppression in the third term with the number of source tasks due to the *i.i.d* assumption about tasks. This implies that it is only possible to get useful bounds for realistic problems when the number of source tasks is large. *Thus, we decided to develop an alternate approach to developing transfer learning bounds.*

3.2.2 Geometric Formulation for Transfer Learning

As our other approach, we developed a ‘geometric’ formulation of transfer learning. In order to obtain valuable mathematical insights without getting mired by the complicated structure of real world data, we decided to first study and understand simplified setups. The motivation of the study was twofold. Given a collection of source tasks and a particular target task:

- Capture the notion of source task relevance for a given target task.
- Capture the complementarity of source task information and the choice & labels of labeled samples of the target task.

3.2.2.1 Results for Linear Gaussian Case

The setup we were initially concerned with was binary classification with linear decision rules of the form $v^T x > 0$ or $v^T x < 0$ for the two different hypotheses. We will assume without loss of generality that the data vector x is k -dimensional and that the covariance of x has been normalized to unity. The above decision rule implicitly means that the two binary hypotheses (H_1 & H_{-1}) are equally likely and the probabilistic model satisfies:

$$p(x|H_1) = N(v_0, I; x); p(x|H_{-1}) = N(-v_0, I; x), \quad (8)$$

where v_0 stands for the (k -dimensional) optimal linear functional for the binary classification problem. Training a learning algorithm on the labeled samples from the target task will yield a linear functional v that is different from v_0 in general. The goal then is to compute an upper bound on the excess risk:

$$q_e - p_e = \Pr(v^T x > 0 | H_{-1}) - \Pr(v_0^T x > 0 | H_{-1}) \quad (9)$$

Denoting the *c.d.f* of the standard Gaussian as $\Phi(\cdot)$, standard manipulations yield the excess risk as:

$$q_e - p_e = \Phi(-\alpha_v ||v_0||) - \Phi(-||v_0||), \quad (10)$$

where $\alpha_v := \cos(\theta_v) = \frac{v^T v_0}{||v|| ||v_0||}$. Therefore, an upper bound on the (worst case) excess risk can be obtained by minimizing the parameter α , or equivalently, maximizing the angle, i.e.

$$q_e - p_e \leq \Phi(-\alpha ||v_0||) - \Phi(-||v_0||) \quad (11)$$

Treating the RHS as a function of $u := ||v_0||$, one can show that the RHS has a maximum at $u_* = \left(\frac{2 \log(\alpha_v)}{\alpha_v^2 - 1}\right)^{1/2}$. Thus, we have:

$$q_e - p_e \leq \Phi(-\alpha_v^{min} u_*(\alpha_v^{min})) - \Phi(-u_*(\alpha_v^{min})) \quad (12)$$

The goal then is to estimate α_v^{min} , leveraging information from the source tasks and the labeled samples from the target task. This can be done as follows. Given a collection of source tasks, we will first assume that the optimal linear functionals have been accurately determined (it is assumed

that there are many labeled samples to learn from for the source tasks) and construct a cone from these. Then, we perform the following steps to estimate α_v^{min} :

1) “Thickening”: Each of these source vectors is thickened followed by a convexification step that gives rise to a polyhedral source cone. This can be thought of as capturing both the uncertainty in the specification of the source vectors as well as a measure of the ‘relevance’ of the source tasks to the target task. For example, if the thickening parameter (call it $\cos \phi$) is small, this would imply that the source tasks are more relevant to the target task and vice versa.

2) “Chopping”: The thickened source cone is reduced from above using data from the labeled samples from the target task. This procedure still yields a polyhedral cone, and the task is to compute the extreme rays of this polyhedral cone (which is a well-studied problem in linear programming), and then compute the maximum angle between pairs of extreme rays.

Finally, the above procedure provides a bound for a given target sample (and collection of source tasks). Ultimately, we would like to convert this to a high probability bound that is valid for all choices of target samples of a given size. It should be possible to come up with a high probability bound of this type using bounded difference inequalities.

We carried out a few experiments on synthetic datasets to validate the bound above.

Experiment 1: For this experiment, we will assume that we have full knowledge of the source task linear functionals. This is a simplified case that can be used to understand the transfer learning algorithm. For this experiment, we also will only be doing chopping from the data vectors, not using the empirical mean to further limit the transfer cone. We will start by specifying the target linear functional v_0 , as well as a set of source task linear functionals $\{v_i | 1 \leq i \leq S\}$ that define an initial transfer cone. The parameters of the experiment are:

- Feature dimension (d) - 6
- Number of source tasks (S) - 2
- thickening radius (ϵ) - 0.01
- Labeled target training sizes (m) - 10, 20, 50, 100, & 200

The *thickening* and *chopping* procedures provide the following bound on the excess risk:

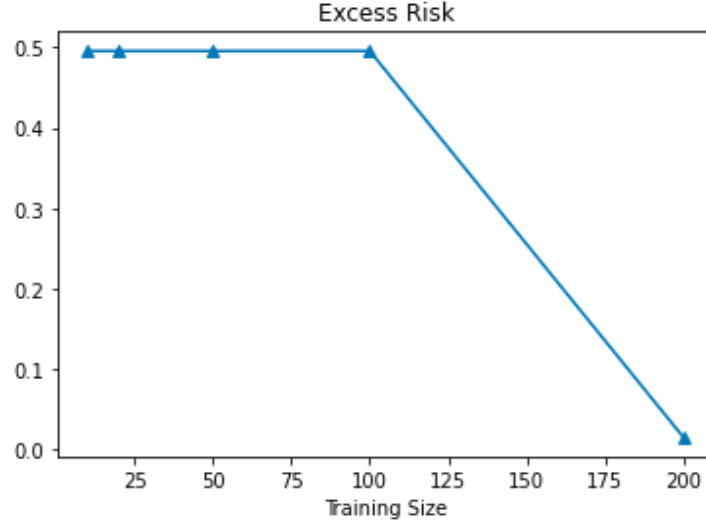


Figure 8: Excess Transfer Risk Bound for Experiment 1

Experiment 2: For this experiment, we want to use the mean of the data to gain additional information about the target task. For this experiment, we start off by repeating the same steps as experiment 1 in data generation, thickening, and chopping. Then, we want to use the learned linear functional:

$$v = \frac{1}{m} \sum_{i=1}^m z_i \quad (13)$$

to construct an L_1 -ball around the mean vector such that it contains the true target linear functional with high probability. We know there is a high probability the learned vector will be within three standard deviations of the true vector. However, this property is with respect to the L_2 -norm, and we want to use the L_1 - norm to ensure everything is polyhedral. This creates an issue: the L_1 - ball is a polyhedron inscribed within the L_2 -ball with the same radius. This means we can get a point that is within 3 standard deviations of the true value but is also outside the ball. We fix this issue by instead taking the circumscribing L_1 - ball, which requires us to increase the radius by a factor of d , where d is the feature dimension. The transfer cone obtained by chopping is then projected onto an L_1 -ball around the mean vector defined by:

$$\|u - v\|_1 \leq \frac{3\sqrt{d}}{\sqrt{m}} \quad (14)$$

The parameters of this experiment are:

- Feature dimension (d) - 15
- Number of source tasks (S) - 2
- thickening radius (ϵ) - 0.01
- Labeled target training sizes (m) - 10, 20, 50, 100, & 200

We compute two versions of the bound – i) with knowledge of $\|v_0\|$ —the true target linear functional, ii) and with no knowledge of $\|v_0\|$, as seen in the figure below.

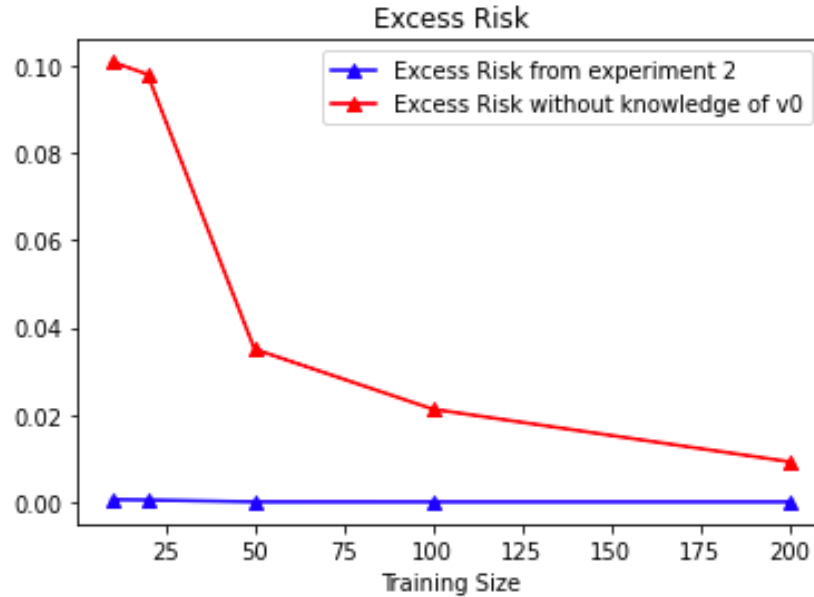


Figure 9: Excess Transfer Risk Bound for Experiment 2

These results were quite encouraging and provided validity to the theoretical ideas developed earlier – at least in the simplified setting. Going forward, the main goal was to answer the following question:

- How does one generalize the above bounds to more *realistic* and *high-dimensional* datasets (e.g., CIFAR100, ImageNet, etc.) where the conditional features for positive and negative samples do not obey a Gaussian distribution?

3.2.2.2 Results for the General Case

One of the issues we encountered when trying to generalize our methods from the simple linear Gaussian case to the general high-dimensional case, was in the *chopping* procedure initiated by adding a set of labeled data points. We start by partitioning the existing generators into a set of size n , which are in the chop, and a set of size m , which are not in the chop. This means that we keep the n generators in the updated transfer cone. Additionally, we add new generators via the projection method. There are $m \cdot n$ such pairs to project, giving a total of $(m + 1) \cdot n$ vectors for the new transfer cone. For our previous experiments with low-dimensional datasets, there are initially $2dS = O(10)$ vectors in the initial transfer cone. After a small number of chops, this results in a cone with $O(10000)$ generators in the final transfer cone. On the other hand, for high-dimensional datasets (as is common in many realistic cases), we start with $2dS > O(100)$ generators. After a single chop, eliminating 600 generators, we have a new transfer cone with $\sim 100K$ generators. This will undergo further chops, which is computationally intractable due to time and memory constraints.

To alleviate this issue, we developed an alternative procedure that scales much more efficiently.

- **Step 1:** Form disjoint partition of $S = S_i \cup S_o$, where:
 - Each s_n in S_i has non-negative dot product with all of the z_j
 - Each s_n in S_o has negative dot product with at least one of the z_j
- **Step 2:** Expansion with L_1 – balls around each source vector
 - For each s_n in S_i place minimum-size L_1 – ball around it so that at least one of the “corners” has zero dot product with at least one of the z_j
 - For each s_n in S_o place minimum-size L_1 – ball around it so that at least one of the “corners” has non-negative dot product with *all* of the z_j
- **Step 3: Variant A**
 - Form the source polytope using the expanded L_1 – balls as described in Step 2.
 - Go to Step 4
- **Step 3: Variant B**
 - Form the source polytope using:
 - For inliers, i.e., elements of S_i , use the expanded L_1 – balls as described in Step 2
 - For outliers, use *only* the corners of the L_1 – balls that have non-negative dot product with *all* of the z_j
 - Go to Step 4
- **Step 4:** Intersect the source polytope with the polytope formed from the 3-sigma L_1 – ball around the mean estimate of the target vector obtained from the labeled samples.

As there isn’t any cutting, any of these procedures should avoid the explosion of facets that cutting yields. Note that the difference between Variants *A* and *B* is that *B* reflects the fact that each of the outlier source vectors doesn’t appear to be as relevant to the target task as the inliers, as each outlier would give the incorrect label to at least one of the labeled samples. Keeping the entire L_1 – balls around the outliers keeps a potentially large region that is equally or even *less* relevant to the target task. Note that Variant *B* In principle may yield only a single corner vector and hence can be too thin (indeed, one can imagine situations with enough labeled samples in which *all* of the source vectors are outliers, as each may have negative dot product with one or another of the labeled samples). We performed various experiments as described below.

Experiment on Synthetic Dataset with updated Algorithm

The parameters of the experiment are:

- Feature dimension (d) - 25
- Number of source tasks (S) - 2
- Labeled target training sizes (m) - 10, 20, 50, 100, & 200

Until now, we have not utilized any source task information in learning the target classifier. We have been using the empirical mean of the target data as the learned target classifier. This essentially correspond to the Maximal likelihood estimate (MLE) of the target data:

$$\widehat{v_0^{MLE}} = \frac{1}{m} \sum_{k=1}^m z_k \quad (15)$$

In principle, we can use a Bayesian approach to utilize the source information as a prior in learning the target classifier and find $\widehat{v_0}$ as the MAP estimate. With the above procedure, we can hope to see the difference in the empirical risk on a test set (test risk) for the two learned classifiers (with and without transfer learning) and compare each of them with the (excess) risk upper bounds. By performing this exercise, we find the results below:

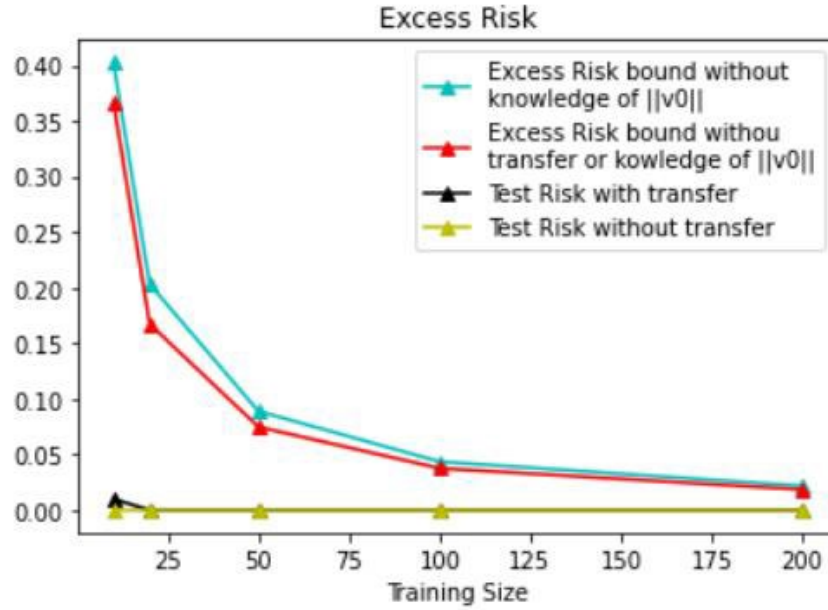


Figure 10: Test risks are compared with the Risk bounds for cases - i) with & ii) without transfer learning

The figure above shows encouraging and expected results:

- The risk upper bound with transfer learning (red curve) is smaller than the excess risk upper bound without transfer learning (cyan curve)
- The actual test risk of the learned classifier for both with and without transfer learning are lower than the risk bounds.
- Finally, we can also see that the test risk of the learned classifier with transfer learning (using the Bayesian approach) is lower than that of the learned classifier without transfer learning (as seen from the **light green** and **black** curves). The black curve is above the light green curve only for the first checkpoint (with ~ 10 samples), but it shows the benefit of using the MAP approach vs the MLE approach. In principle, the benefit is more pronounced in higher dimensions. However, in those cases it turns out that the *risk bounds* are not computable due to the cones being obtuse rather than acute – a purely technical road-block rather than a conceptual one. Therefore, we only use a problem with $d=25$ dimensions to be able to show all the curves.

Experiment on MNIST

For this experiment, we divided the MNIST data and split it into four source tasks and a target task, all of which are binary classification. We then normalized the dataset for each task independently, so that each has mean zero and identity covariance.

We created a model for each source task of the form $y = a_i^T B x$, where B is a shared orthogonal representation across all four source tasks and a_i is a task-specific classifier. We then calculated a representation of the target task data $\tilde{x} = B x$. Note that the representation should still have mean zero and identity covariance due to the orthogonality of the representation. After this exercise, the normalized MNIST data with the shared representation has the following parameters:

- Feature dimension (d) - 100
- Number of source tasks (S) - 4

The empirical data distribution obtained from the above procedure for MNIST closely approximates a Gaussian distribution. However, there are some additional assumptions that are implicit in the bound computed for the excess transfer risk. To state these assumptions, let us denote the conditional distribution of the input samples (in representation space) for each hypothesis (H_1/H_{-1}) as:

$$p(x|H_1) = f_1(x) \tag{16}$$

$$p(x|H_{-1}) = f_{-1}(x) \tag{17}$$

The assumptions underlying the bound are:

- The means of the two hypotheses are negative of each other,
- The conditional data distribution under the two hypotheses are reflected versions of each other.
- The covariance of the conditional data distributions is identity.
- The decision rule is based on a linear functional of the feature vector.

Checking for the above assumptions for the MNIST case, it was found that assumption 1) is satisfied, assumption 2) is approximately true (shown in **Figure 11**), while assumption 3) is completely false. Assumption 4) is hard to check a priori. In particular, it was found that the covariances of $f_1(x)$ and $f_{-1}(x)$ are neither equal nor diagonal.

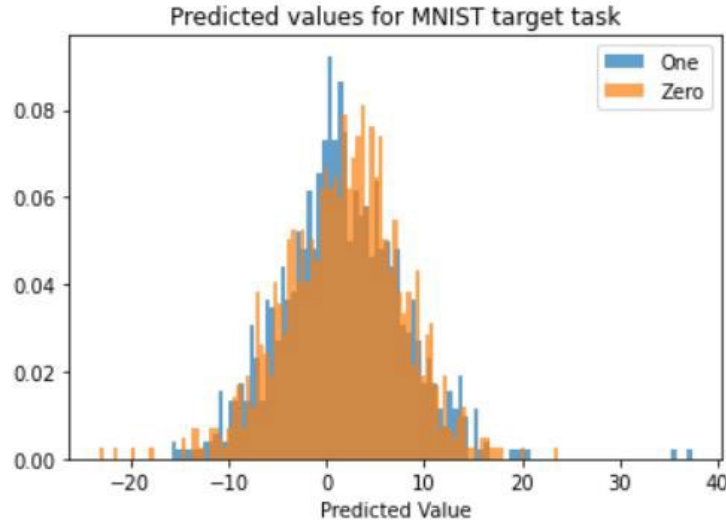


Figure 11: Empirical distributions of $f_1(x)$ and $f_{-1}(x)$ for MNIST

In order to deal with this, we start by normalizing the data to have zero mean. This will not change the marginal variance of the classes but does ensure that the conditional means are negations of each other. We will next apply a linear transformation Q such that both classes have independent components in the transformed space. We can apply the simultaneous diagonalization theorem to find a Q such that one class has an identity covariance and the other has diagonal covariance. This gives us a transformed space given by:

$$U = Q (x - \mu) \quad (18)$$

With the following properties:

$$\mathbb{E}(U|H_1) = \hat{\mu} \quad (19)$$

$$\mathbb{E}(U|H_{-1}) = -\hat{\mu} \quad (20)$$

$$\text{Var}(U|H_1) = \mathbb{I} \quad (21)$$

$$\text{Var}(U|H_{-1}) = D \quad (22)$$

Then we make the following observation: the linear transformation Q means the components U_k of the transformed space are the weighted sum of several random variables. As such, the *Central Limit Theorem* tells us the resulting transformed space should follow a Gaussian distribution. We can look at this for the *cats vs dogs* dataset to see whether empirical results on a test set match this theory and intuition

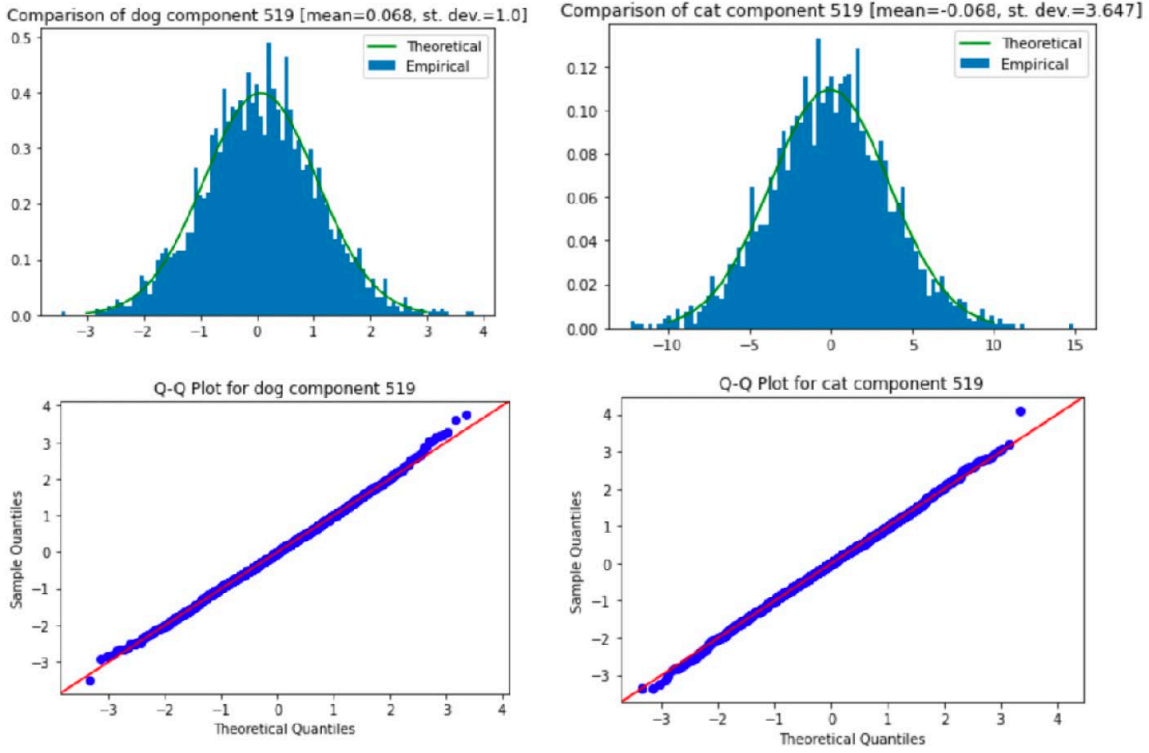


Figure 12: Empirical conditional distribution for each class and Q-Q plots for the quantiles

Figure 12 shows that the conditional distribution for each class indeed follows a Gaussian distribution to a very good approximation. For example, in component 519, the dogs distribution is approximately normal with mean 0.618 and standard deviation 1, whereas the cats distribution is approximately normal with mean -0.618 and standard deviation 3.047. Checking many different dimensions shows similar results, so we have a good method for normalizing our data.

In order to compute bounds on the excess transfer risk, however, we will need to consider how the classifier in the original space (say f) is transformed in the U space. Since the two spaces are connected via the matrix Q , we obtain the following classifier in U space:

$$v = Q^{-T} f \quad (23)$$

Given this, we notice that the optimal test for two hypotheses is a likelihood ratio, which can be expressed in terms of log likelihood as:

$$y = \begin{cases} 1 & \ell(H_1|u) - \ell(H_{-1}|u) > 0 \\ -1 & \ell(H_1|u) - \ell(H_{-1}|u) < 0 \end{cases} \quad (24)$$

Using the log-likelihood expressions for the Gaussian distribution, we get:

$$\ell(H_1|u) - \ell(H_{-1}|u) = \frac{-1}{2} \sum_{k=1}^p \left[\ln d_k + \left(1 - \frac{1}{d_k}\right) \hat{\mu}_k^2 \right] - 2 \sum_{k=1}^p \left(1 - \frac{1}{d_k}\right) \hat{\mu}_k u_k - \frac{1}{2} \sum_{k=1}^p \left(1 - \frac{1}{d_k}\right) u_k^2 \quad (25)$$

Note that the above is a linear function in augmented feature space $\tilde{u}$ in R^{2p} dimensional space with p u -variables and p u^2 variables. Using the above, we can write the excess transfer risk as:

$$P(\text{signum}(\hat{v}^T \tilde{u}) \neq y) - P(\text{signum}(v^{*T} \tilde{u} + c) \neq y) = \frac{1}{2} [\hat{\alpha} + \hat{\beta} - \alpha^* - \beta^*] \quad (26)$$

where $\hat{\alpha}, \hat{\beta}, \alpha^*, \beta^*$ are the type I and type II errors of the learned and optimal classifiers respectively. In particular, the type I error of the learned classifier is given by:

$$\begin{aligned} \hat{\alpha} &= P(v^T u > 0 | H_{-1}) \\ &= \int_{\{v^T u > 0\}} \phi(u | -\hat{\mu}, D) du \end{aligned} \quad (27)$$

Where $\phi(u | -\hat{\mu}, D)$ is the pdf of the conditional input distribution in ' $\tilde{U}$ ' space with mean $-\hat{\mu}$ and diagonal covariance D . In general, it is challenging to estimate this pdf as $\tilde{U}$ space has both linear and quadratic features. The same issue arises with the optimal classifier as well. For this case, we need to estimate the distribution of the quantity $v^{*T} u^e | H_{-1}$, where the components in u^e are not independent. To this end, we use the Berry-Esseen Theorem for the quantity that applies to a sum of independent random variables, which may not be from the same distribution. Based on this, we construct the quantity Z_i , given by:

$$Z_i = v_i^* U_i + v_{i+p}^* U_i^2 \quad (28)$$

Since $U_i; i = 1, 2, \dots, p$ are independent, so is Z_i , with $v^{*T} u^e = \sum_i Z_i$. The Beery-Esseen theorem allows one to bound the *cdf* of the distribution of $v^{*T} u^e | H_{-1}$ in terms of a standard Gaussian *cdf*. Then the Type I error of the optimal classifier are given by:

$$\alpha^* \geq \Phi \left(\frac{-v^{*T} \mathbb{E}(U^{(e)} | H_{-1}) - c}{\sqrt{v^{*T} \text{Var}(U^{(e)} | H_{-1}) v^*}} \right) - \frac{0.7915 r_n}{(v^{*T} \text{Var}(U^{(e)} | H_{-1}) v^*)^{\frac{3}{2}}} \quad (29)$$

and similarly for Type II error. Collecting all expressions together, we get the final result for the excess transfer risk bound:

$$\begin{aligned} &\leq \Phi \left(\frac{-\|v_A\| \|\mu_{UA}\| \cos \theta}{\sqrt{v^T D v}} \right) + \Phi \left(\frac{-\|v_A\| \|\mu_{UA}\| \cos \theta}{\sqrt{v^T v}} \right) \\ &- \Phi \left(\frac{-v_A^{*T} \mathbb{E}(U_A^{(e)} | H_{-1})}{\sqrt{v^{*T} \text{Var}(U^{(e)} | H_{-1}) v^*}} \right) - \Phi \left(\frac{-v_A^{*T} \mathbb{E}(U_A^{(e)} | H_1)}{\sqrt{v^{*T} \text{Var}(U^{(e)} | H_1) v^*}} \right) \\ &+ \frac{0.7915 r_n}{(v^{*T} \text{Var}(U^{(e)} | H_{-1}) v^*)^{\frac{3}{2}}} + \frac{0.7915 s_n}{(v^{*T} \text{Var}(U^{(e)} | H_1) v^*)^{\frac{3}{2}}} \end{aligned} \quad (30)$$

Excess transfer risk <=

The main upshot of our approach is that the bound for the excess transfer risk can be computed in terms of *cdf* of the standard Gaussian with computable parameters and computable corrections to it.

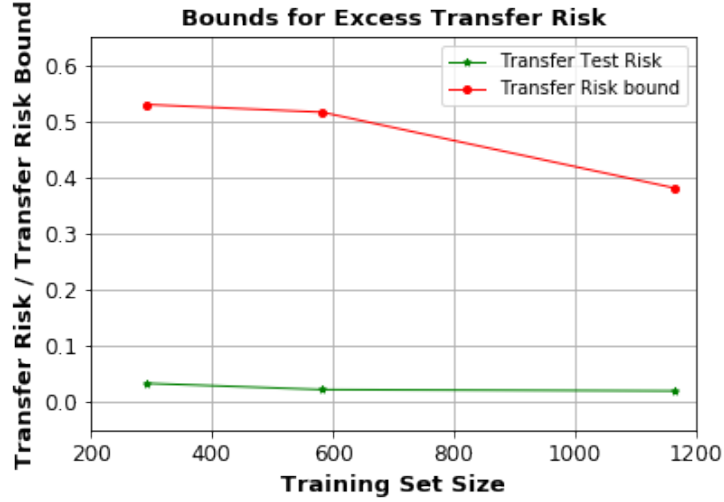


Figure 13: Excess transfer risk bounds with the Imagenet shared representation

In **Figure 13**, we show preliminary results for validating our formalism for the excess transfer risk bounds. The details of the experiment are as follows:

- **Shared representation:** Xception architecture trained on Imagenet dataset
- **Target task:** Cats vs Dogs binary classification (Kaggle dataset)
- **Source tasks:** a) Dogs vs Fish dataset (Imagenet), b) Cats vs Horses dataset (CIFAR-10)
- **Workflow:**
 - Perform simultaneous diagonalization & estimate parameters in U variables (using pseudo-labels): $\{\mu_A, D, \mathbf{E}[U_A^e|H_1], \mathbf{E}[U_A^e|H_{-1}], \mathbf{Var}[U_A^e|H_1], \mathbf{Var}[U_A^e|H_{-1}]\}$
 - Estimate transfer cone antipodal angle by intersecting thickened target cone with that of thickened source cone.
 - Compute transfer risk bound by estimating $\{\alpha, \beta, \alpha^*, \beta^*\}$
 - Compare bound with test risk of learned classifier on target task that is fine-tuned from source classifiers.

We see that our formalism provides a non-vacuous upper bound on the transfer risk for real-world datasets and tasks. However, while this bound is non-vacuous, it is still quite loose. The first two terms in the bound are the most important, in particular the angle θ between the optimal classifier and the learned classifier. The reason our previous bound was loose was due to the fact that we used the worst case bound on $\cos\theta$. Therefore, we decided to use a sampling based estimate of the bound.

For our new approach, we look at combining the bagging and sampling approaches into a single method. The idea behind this method is to use the following process:

1. Shuffle the dataset into a new order.
2. Train a classifier on the shuffled dataset.
3. Create cone that is compatible with source vectors - the source cone.
4. Intersect L^1 ball around the learned classifier with the source cone to generate the *target cone*.
5. Sample M ($= 30$) vectors from the target cone.
6. Compute the bound on excess risk for each sampled angle.
7. Repeat process for K ($= 5$) times for 5 different cuts of the dataset, and we get a total of $M * K$ ($= 150$) sampled values for the excess transfer risk bound.
8. Take the maximum of these values as the bound for the excess risk.

Using the above approach, we were able to achieve a much tighter bound on the excess transfer risk for the *cats vs dogs* dataset

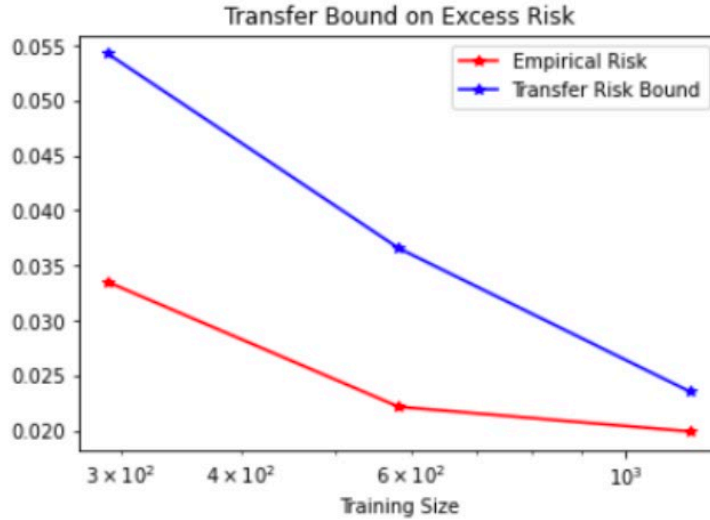


Figure 14: Bound obtained with the sampling approach.

This gives us a fairly tight bound, but it is not completely precise as there is still a small probability that the true bound is larger than the above sampling based bound.

We can also develop a precise high probability upper bound by leveraging Chebyshev's inequality:

$$\begin{aligned}
 & \Pr_{v^* \sim \text{Uni}(TC); \mathbf{Z}_m \sim D^m} \left[q_e - p_e \leq \mathbb{E}_{v^* \sim \text{Uni}(TC)} \left(\hat{\mathbb{E}}_{\mathbf{Z}_m \sim D^m} [B(v^*, \mathbf{Z}_m)] + 2\hat{R}_m(\mathcal{B}) \right) + 3 \sqrt{\frac{\log(2/\delta_2)}{m}} + \frac{1}{2\sqrt{\delta_3}} + \frac{1}{2\sqrt{\delta_1}} \right] \\
 & \geq 1 - \delta_3 - \delta_2 - \delta_1
 \end{aligned} \tag{7}$$

where $B(v^*, \mathbf{Z}_m)$ denotes the RHS of the bound on the previous page. Note that this final bound depends on the Rademacher complexity of the bound function class $\mathcal{B}$. The expectation over

$Uni(TC)$ can be presumably reliably estimated by an empirical mean over a large number (N) of samples from the transfer cone TC with minimal error. So, the above bound provides formal grounding to the procedure described earlier and should provide a computable and precise probabilistic bound for the excess transfer risk. As mentioned earlier, it does require one to estimate the empirical Rademacher complexity of the function class B , which is not clear how to do. Also, if this complexity is large (remains to be seen), it can give rise to a loose bound. Another option may be to make some additional assumptions about the bound function $B(v^*, Z_m)$ to come up with a tighter bound.

At present, we are working on a paper to systematically present and publish our results [].

3.3 Understanding of Structured Prediction and Multi-class classification with Constraints

During the program, we initiated collaboration with Prof. Dan Roth's group at U Penn on the topic of *structured prediction* and started exploring a number of potential new approaches in this direction along with our subcontractor – Prof. Broderick's group at MIT.

Developing theoretical bounds on structured prediction is a challenging problem. The general problem can be decomposed into three sub-problems:

- An effective way to encode constraints (arising from the underlying structure) into the model has to be developed. There are two available choices: i) Scoring function, ii) Loss function.
- A way to describe the dependency for *non-iid* data and to derive concentration bounds for such data should be developed. Examples include: i) mixing coefficient as in [15], Dobrushin's condition, etc.
- A measure of complexity for the model class should be estimated. Two natural choices are: i) Extension of Rademacher complexity to factor graphs that also takes structure constraints into account, and ii) PAC-Bayesian methods that use KL-divergence between prior and posterior as a measure of complexity.

We are focusing on the first problem for now as it will decide how we proceed with the other two problems. For the first sub-problem above, we are currently exploring the two options mentioned:

i) *Encoding constraints into the Loss Function*

A natural way to encode constraint is to add a term in the loss function:

$$l_c(h, x, y) = l(h(x), y) + \lambda d_c(x, h(x)), \quad (31)$$

where d_c measures the violation of constraints and λ is the tradeoff hyperparameter. We can view $d_c(x, y')$ as a regularization function. One way to analyze the benefit of regularization is to consider the stability of the learning algorithm A . In particular, for the structured prediction problem, we may consider a new way to perturb the data Z . For example, we can delete (or replace) one component z_i in a single example. Denoting the perturbed (partial) example as $Z^{(i)}$, one can train the model with and evaluate its performance on the masked token, i.e. compare:

$$E[L(A(Z^{(i)}), z_i) - L(A(Z), z_i)] \quad (32)$$

Intuitively, this difference will be smaller in the case with constraints since the information of z_i can be partly recovered by the other components of the structure. This would imply that the algorithm in the presence of constraints is more stable than without. We are currently trying to formalize this idea by proving some inequalities by leveraging ideas from the paper of London et al [15].

ii) Encoding constraints into the Scoring Function

An alternative method is to encode the constraint directly into the scoring function (classifier), since it enables us to increase the optimal performance of the model. This can be readily done for linear models. For example, the Constraint Conditional Model (CCM) augments the scoring function as:

$$s_w(x, y) = \theta^T \Phi(x, y) - \lambda d_c(x, y) \quad (33)$$

where θ is the weight, $\Phi(x, y)$ is the feature vector, $d_c(x, y)$ measures the violation of constraints and λ is the tradeoff hyperparameter. It is unclear how to generalize the above for more general models, in particular deep neural networks.

If we proceed in this direction, a good complexity measure is the *factor graph Rademacher complexity* (a generalized Rademacher complexity) proposed in Cortes et al [16].

$$\hat{R}_S^G(H) = \frac{1}{m} E_\epsilon [\sup_{h \in H} \sum_{i=1}^M \sum_{f \in F_i} \sum_{y \in Y_f} \sqrt{|F_i|} \epsilon_{i,f,y} h_f(x, y)] \quad (34)$$

However, we would need to improve this result to allow the generalization bound to depend on the *size* of the structure. Then the model can generalize from even one example, just as discussed in London et al [15]. It may be possible to leverage ideas in [17] to incorporate constraints into the Rademacher complexity, which we are currently looking into.

3.3.1 Structured Prediction with Partial Labels and Constraints

In Phase 1, we explored the realistic problem of deriving bounds for structured prediction tasks where training data is only partially labeled.

The first setup we consider is the case where one has partial labels with no additional constraints (side information). In this setup, we have studied the case where the partial labels obey the following model. The input (covariate) space is $\mathbf{X}$, the output space is $\mathbf{Y}$. Each labeled example consists of the covariate part and an n -part label vector: $Z_j = \{X_j, \{Y_j^{(i)}\}\}, i = 1, 2, \dots, n$. We only have access to a partially labelled $\tilde{Z}_j$ where the revealing mechanism for some known $\rho \in [0, 1]$ is as follows. We always observe the full covariate X_j , but we only observe the labels $\tilde{Y}_j$ as below:

$$\begin{aligned} V_j^{(i)} &\sim \text{Ber}(\rho) \\ \tilde{Y}_j &= \{Y_j^{(i)}, : V_j^{(i)} = 1\} \end{aligned} \quad (35)$$

In this case, we are able to show that if the labelling rate is ρ , then effectively there are $M\rho$ samples used for learning. For the case when the hypothesis space is finite, one can derive a generalization bound for this case:

Proposition 2 (First learning theorem). *If we define the empirical risk minimizer*

$$\hat{h} := \arg \min_h \frac{1}{M} \sum_{m=1}^M \tilde{L}(h, \tilde{Z}_m) \quad (36)$$

then with probability at least $1 - \delta$ over $\{\tilde{Z}_j\}_{j=1}^M$

$$\mathbb{E}_Z L(\hat{h}, Z) \leq \min_h \mathbb{E}_Z L(h, Z) + \frac{\log(|\mathcal{H}|/\delta)}{M\rho} + \sqrt{\frac{\log(|\mathcal{H}|/\delta)}{M\rho}}. \quad (37)$$

There are various avenues for refining and generalizing the above result. For example, what happens when each covariate has a separate labelling rate ρ_i ? Another refinement is the development of bounds for infinite hypothesis classes using more advanced complexity measures like Rademacher complexity.

We have also studied an extension of the above setup – structured prediction with partial labels and deterministic constraints (side information). In particular, we study the case where for each example $\tilde{Z}_j$, we have a constraint set $S(X_j)$ that depends on X_j , i.e. the tokens in a sentence. $S(X_j)$ tells us which label components are known to be equal to each other given X_j :

$$S(X_j) = \{(p, q) : Y_j^{(p)} = Y_j^{(q)}\} \quad (38)$$

We then introduce indicators $T_j^{(i)}$ that depend on the revealing variables $V_j^{(i)}$ of the j^{th} example and the constraint set $S(X_j)$. Essentially, $T_j^{(i)} = 1$ either when $Y_j^{(i)}$ is observed or if it can be inferred from the other labels and constraint $S(X_j)$. Using this, we can construct an unbiased estimator of the true risk. Then, we can derive a generalization bound for this case with a finite hypothesis class:

Proposition 5 (Variant of second learning theorem). *If we define the empirical risk minimizer*

$$\hat{h} := \arg \min_h \frac{1}{M} \sum_{m=1}^M \tilde{L}_2(h, \tilde{Z}_m) \quad (39)$$

then with probability at least $1 - \delta$ over $\{\tilde{Z}_j\}_{j=1}^M$

$$\mathbb{E}_Z L(\hat{h}, Z) \leq \min_h \mathbb{E}_Z L(h, Z) + B \log(|\mathcal{H}|/\delta) + \sqrt{W \log(|\mathcal{H}|/\delta)} + \sqrt{\frac{\log(|\mathcal{H}|/\delta)}{M}}, \quad (40)$$

where B and W are as defined in Lemma 4

(41)

$$\begin{aligned}
B &:= \frac{1}{M} \max_{1 \leq j \leq M} \left(\frac{1}{n} \sum_{i=1}^n \frac{1}{\Pr(T_j^i = 1)} \right) \\
W &:= \frac{1}{M^2} \sum_{j=1}^M \left(\frac{1}{n^2} \sum_{p,q} \frac{\mathbb{E}(T_j^p T_j^q)}{\Pr(T_j^p = 1) \Pr(T_j^q = 1)} \right).
\end{aligned} \tag{42}$$

A natural extension of the above case (as before) is to develop bounds for infinite hypothesis classes using Rademacher complexity with constraints. Another extension is to develop learning theory bounds for more realistic constraints such as probabilistic constraints (side information).

This would be very useful because many constraints in NLP tasks are of this form. Another thing to explore is how to resolve potential conflicts with probabilistic side information? For example, there could be two events with high probability, like $\{X1 = X2 \text{ with probability } 0.9\}$ and $\{X1 = X3 \text{ with probability } 0.8\}$, but in the data, $X1$ is not observed, and the observed $X2$ is not equal to the observed $X3$. Finally, it would also be very useful to incorporate constraints of the form “if two labels are this and this, then the third label must be this”? These types of constraints arise in the NLP problem of *temporal extraction*.

3.3.2 Generalization Bounds for Constrained Models

In phase II, our goal was to generalize the work in phase I and study the following cases: i) Realizable hypothesis with noise-free constraint, ii) Realizable hypothesis with noisy constraint, iii) non-realizable hypothesis with noise-free constraint, and iv) non-realizable hypothesis with noisy constraint. For the first case, we were able to prove a generalization bound. The details are as follows.

We first define some notation:

- $v : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$ be the violation function associated to the constraint C (typically $v(x, y) = 1\{y \notin C(x)\}$)
- $V(f) = \mathbb{E}[v(x, f(x))]$ be the expected violation of a hypothesis $f : \mathcal{X} \rightarrow \mathcal{Y}$
- $\hat{V}(f) = \sum_{x \in S_U} [v(x, f(x))]$ be the empirical violation given an unlabeled dataset S_U (of size m_U)
- $L : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$ be the classification error: $L(x, y) = 1\{y \neq f_{\text{oracle}}(x)\}$ where f_{oracle} is the true classifier.
- $R(f)$ be the risk $\mathbb{E}[L(x, f(x))]$
- $\hat{R}(f)$ be the empirical risk given a labeled dataset S_L (of size m_L)

Further, we denote:

$$\mathcal{F}_a^b := \{f \in \mathcal{F} | a \leq V(f) \leq b\} \quad \hat{\mathcal{F}}_a^b := \{f \in \mathcal{F} | a \leq \hat{V}(f) \leq b\} \tag{43}$$

where the second is the empirical version of the first. Given m_U unlabeled data, we only consider hypothesis with zero empirical violation since the oracle (for the realizable hypothesis) has zero violation by definition. That is, we find the risk minimizer in $\hat{\mathcal{F}}_0^0$. Now, given $\hat{f}_0^*$ to be the ERM minimizer, we can use standard generalization bounds to bound the risk of the ERM classifier with probability greater than $1 - \frac{\delta}{2}$:

$$(44)$$

$$R(\hat{f}_0^*) - R(f_{\text{oracle}}) = R(\hat{f}_0^*) \leq B_M(\delta/2, m_L, \hat{\mathcal{F}}_0^0, L)$$

where $B_M(\delta, m, F, l) := B(\delta, m, F, l) + \sqrt{\frac{\log(\frac{4}{\delta})}{2m}}$. Second, since $V(f)$ is also a loss function by assumption, we can also write the following bound:

$$\sup_{f \in \mathcal{F}} V(f) - \hat{V}(f) \leq B(\delta/2, m_U, \mathcal{F}, v) \quad (45)$$

This means that with probability greater than $1 - \frac{\delta}{2}$, each hypothesis f in $\hat{V}(f) = 0$ will have an expected violation of at most $B(\delta/2, m_U, F, v)$. This means that $\widehat{\mathbf{F}}_0^0$ is contained in $\mathbf{F}_0^{B(\delta/2, m_U, F, v)}$. Therefore, by the union bound, with probability at least $1 - \delta$, we have:

$$R(\hat{f}_0^*) \leq B_M(\delta/2, m_L, \hat{\mathcal{F}}_0^0, L) \leq B_M(\delta/2, m_L, \mathcal{F}_0^{B(\delta/2, m_U, \mathcal{F}, v)}, L) \quad (46)$$

When one has a large amount of unlabeled data, $m_U \rightarrow \infty$, and the benefit of the constraints depends on the size of $\widehat{\mathbf{F}}_0^0$ relative to $\mathbf{F}$.

The above result is valid for learning with constraints (encoding constraints in the loss function), which should be contrasted with the case for inference with constraints (encoding constraints as a model feature). We developed a systematic formalism to study the two cases.

Starting with ‘Inference with constraints’, we studied the case with noiseless ‘hard’ constraints. In this case, with a base model, we make constrained prediction. This provides better performance than the base model (with no constraints) as the hypothesis is improved. We have developed theoretical results to show that the empirical error in this case is smaller while the complexity term is the same.

For ‘Learning with constraints’, we study the case with two assumptions: i) the optimal hypothesis satisfies constraints, and ii) the constraints are noiseless. Learning with constraints provides faster learning where one can reach higher performance quickly compared to the base model without constraints. The asymptotic performance of the final model with constraints and without constraints will be the same since the hypothesis space is the same. Again, we developed theoretical results showing that learning with constraints leads to the same empirical error with a smaller complexity term.

We carried out some experiments on the Named Entity Recognition (NER) task with ‘BIO’ constraints. In particular, the constraint is that the ‘O’ label cannot be followed by the ‘I’ label for any set of entities. A neural network model was trained on this task for all three cases: i) base model (without constraints), ii) learning with constraints, and iii) inference with constraints. The results are shown in **Figure 15**.

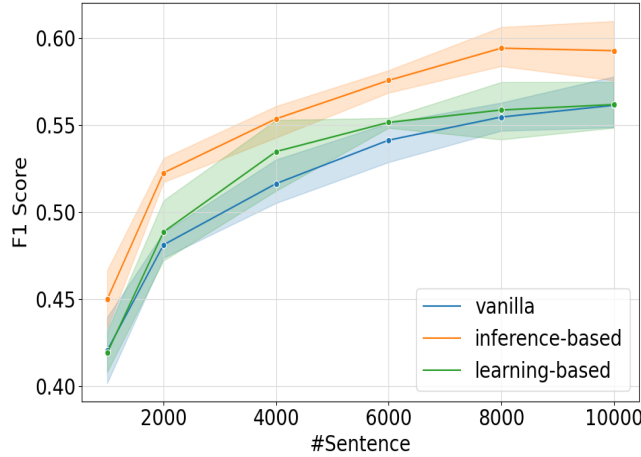


Figure 15: Performance on NER task trained with Neural Network Model

Going forward, we would like to better compare the learning method and the inference method in a quantitative manner with multiple experiments. The goal is to better understand the tradeoffs between the two methods. At present, we have found that for models with low/moderate capacity and noiseless constraints, inference-based methods seem to be better. More experiments are needed to confirm this.

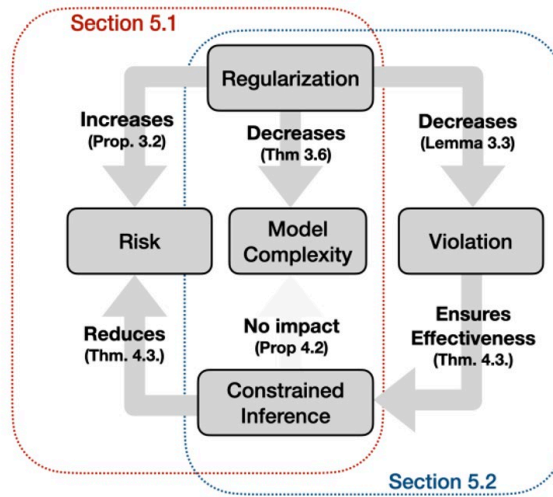


Figure 16: Illustration of the results developed in [18]

Our main result [18] showing the tradeoff between the two approaches can be summarized as follows. Prior knowledge and symbolic rules in machine learning are often expressed in the form of label constraints, especially in structured prediction problems. We compared two common strategies for encoding label constraints in a learning system, regularization with constraints and constrained inference, by quantifying their impact on the model performance. For the regularization approach, we showed that it narrows the generalization gap by precluding models that are

inconsistent with the constraints. However, its preference for small violations introduces a bias toward a suboptimal model. For constrained inference, we showed that it reduces the average loss by correcting a model’s violation, and hence turns the violation into an advantage. However, it has not effect on the generalization gap (the Rademacher complexity remains the same) . Given these differences, we further explored the use of two approaches together and propose conditions for constrained inference to compensate for the bias introduced by regularization, in order to improve both the model complexity and the optimal risk. We found that under certain conditions, this suggests the possibility for CCM to reduce the additional risk introduced by regularization. **Figure 16** provides an illustration of the above statements.

4 RESULTS AND DISCUSSION

This section provides a summary of *Learning with Optimal Labels*’ results for the various tests and evaluations in the three phases. The results have already been described in detail in the previous section.

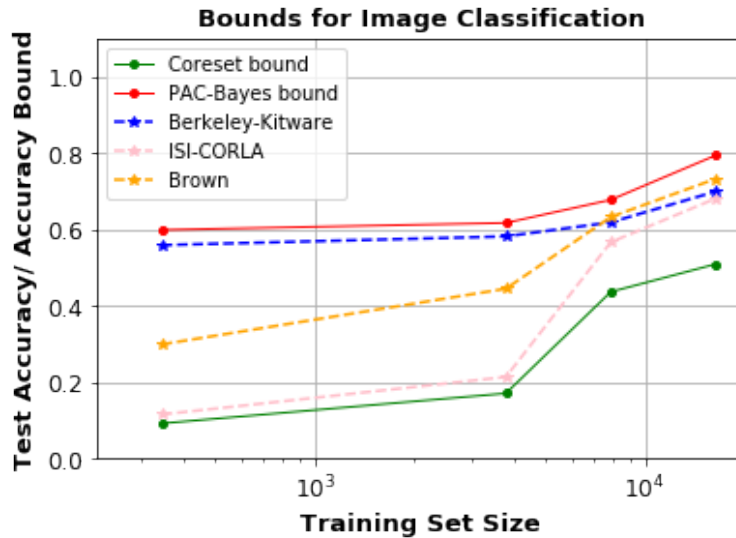


Figure 17: Bounds for Image Classification for 4 TA1 models.

- **ML Task: Image Classification (Evaluation 1)**
- **Dataset: DomainNet-Clipart**
- **Models: ISI, Berkeley-Kitware, Brown University**

We computed upper and lower risk bounds valid for four TA1 models, as shown in **Figure 17**. Our methods reliably show how the models are performing relative to the bounds.

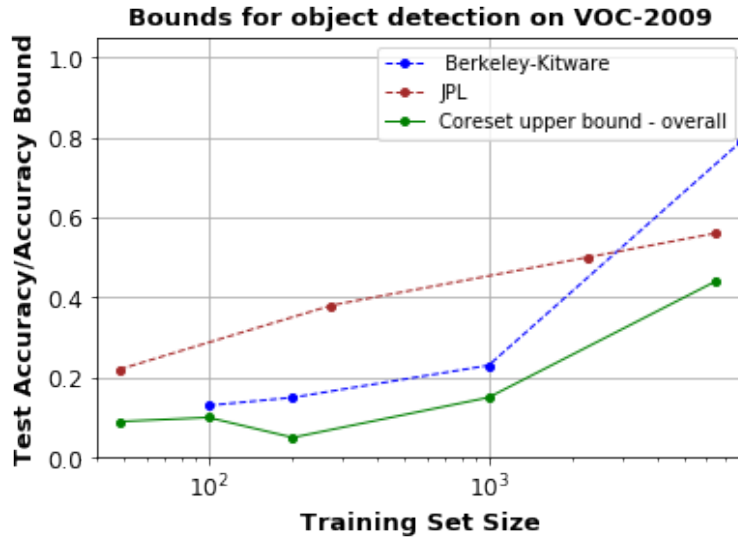


Figure 18: Bounds for Object Detection TAl models.

- ML Task: Object Detection (Evaluation 2)
- Dataset: VOC-2009
- Models: JPL, Berkeley-Kitware

Figure 18 provides lower bounds on the mAP (mean average precision score) for object detection for two TAl models. The upper bounds, although computed, are not shown since they are vacuous (have values above 1.)

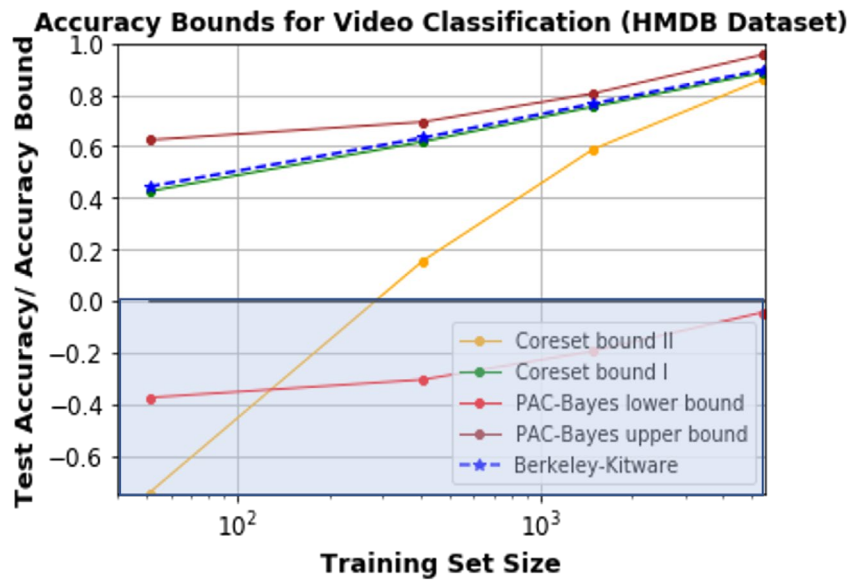


Figure 19: Bounds for Video Classification for Berkeley-Kitware Model.

- **ML Task: Video Classification (Evaluation 3)**
- **Dataset: Human Motion Database (HMDB)**
 - Clips from various movies with 51 labeled actions, e.g., walking, running, chewing, etc.
- **Model: Berkeley-Kitware**

Figure 19 shows results for both upper and lower bounds for video classification for the Berkeley-Kitware model. Many bounds have been shown for completeness even if they are vacuous. We are able to obtain tight lower accuracy bound with Coreset bound I and upper accuracy bound with PAC-Bayesian approach.

We also obtained tight upper bounds for the excess transfer risk using the geometric formulation of transfer learning that we developed.

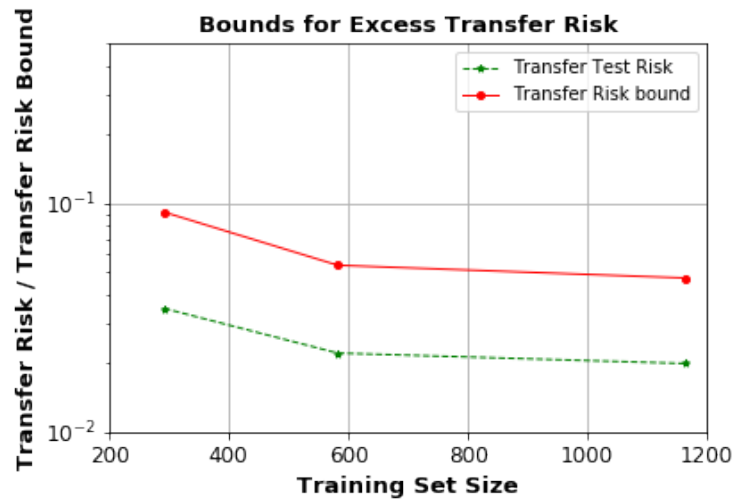


Figure 20: Upper bound on the excess transfer risk with the geometric approach

- **Shared representation:** Xception architecture trained on Imagenet.
- **Target task:** Cats vs Dogs binary classification (Kaggle dataset).
- **Source tasks:**
 - a) Dogs vs Fish dataset (Imagenet)
 - b) Cats vs Horses dataset (CIFAR-10)

The methods developed are applicable generally, subject to shared feature representation assumption, see section 3.2.2.

5 CONCLUSIONS

STR, together with subcontractor MIT and consultants Prof. V. Chandrasekaran and Prof. A. Willsky developed theoretical results for LwLL Task Area 2 with interesting potential applications for its Learning with Optimal Labels (LoL) effort. The key achievements of the STR team were:

- We developed of explicit and useful *lower and upper risk bounds* for ML models (including complex DNNs developed by TA1 performers) for many supervised ML tasks: *Image Classification, Object Detection, and Video Classification*.
- We developed of novel *geometric* formulation of Transfer Learning and construction of novel upper bounds for *excess transfer risk*.
- We developed formalism for understanding precise tradeoff between regularization and inference for *ML tasks with label constraints*.

Our results are generally applicable – they are valid for any ML model including complex deep neural networks. We provided convincing proof of this in the three evaluation tasks over the course of the program. While our results provide significant improvements to the state of the art, we would also like to point out some limitations. *These limitations apply to all current theoretical approaches to machine learning (including ours), so the questions we lay out are open questions for the whole field.*

The main limitation is that all current results are based on Statistical Learning Theory, which has intrinsic limitations for out-of-distribution generalization. Our opinion is that one really needs some neuro-symbolic approach to make seminal advances in leveraging prior out-of-distribution (but semantically related) information. Symbolic information is *key*, it could be logic-based, knowledge-based or physics/dynamics based or a combination of these. Understanding how to combine neural and symbolic approaches will ultimately lead to data/label efficient models (similar to what is done in the human brain).

6 REFERENCES

- [1] Douglas, S., Kumar, P., Prasanth, R.K., “*Risk bounds for learning via Hilbert coresets*”, *JAIR* 2022 (under review), *arXiv 2103.15569*
- [2] Campbell, T., and Broderick, T., *Automated Scalable Bayesian Inference via Hilbert Coresets*, *JMLR* 2019
- [3] Campbell, T., and Broderick, T., *Bayesian Coreset Construction via Greedy Iterative Geodesic Ascent*, *ICML* 2018
- [4] <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.htm>
- [5] <http://komarix.org/ac/ds>
- [6] Rahimi, A., Recht, B., *Random Features for Large-Scale Kernel Machines*, *NIPS* 2007
- [7] Begin, L., Germain, P., Laviolette, F., Roy, J.F., *PAC-Bayesian Bounds based on Renyi Divergence*, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)* 2016
- [8] Banerjee, A., Chen, T., and Zhou, Y., *De-randomized PAC-Bayes Margin Bounds: Applications to Non-convex and Non-smooth Predictors*, *arXiv 2002.09956*
- [9] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “*End-to-end object detection with transformers*,” *CoRR*, vol. abs/2005.12872, 2020.
- [10] K. Chen, J. Li, W. Lin, J. See, J. Wang, L. Duan, Z. Chen, C. He, and J. Zou, “*Towards accurate one stage object detection with ap-loss*,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [11] Baxter, J., *A Model of Inductive Bias Learning*, *Journal of Artificial Intelligence Research* 12 (2000) 149–198
- [12] Mauer, A., Pontil, M., Romera-Paredes, B., *The Benefit of Multi-Task Representation Learning*, *JMLR* 17 (2016) 1-32
- [13] Tripuraneni, N., Jordan, M., Jin, C., *On the Theory of Transfer Learning: The Importance of Task Diversity*, *2006.11650*
- [14] S. Douglas, P. Kumar, V. Chandrasekaran, A. Willsky, “*Geometric Bounds for Transfer Learning*”, Preparing for submission.
- [15] London, B., Huang, B., Getoor, L., *Stability and Generalization in Structured Prediction*, *JMLR* (2017) 16 1-52.
- [16] Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. [Structured prediction theory based on factor graph complexity](#), *Advances in Neural Information Processing Systems (NIPS)* 2016).
- [17] Tulabandhula, T., Rudin, C. *Generalization Bounds for Learning with Linear, Polygonal, Quadratic and Conic Side Knowledge*, *arXiv: 1405.7764*
- [18] K. Wang, H. He, T. Nguyen, P. Kumar, D. Roth, “*On Regularization and Inference with Label Constraints*”, To be published in *ICML* 2023

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

AFRL	Air Force Research Laboratory
BAA	Broad Agency Announcement
CIFAR	Canadian Institute For Advanced Research
DARPA	Defense Advanced Research Projects Agency
DNN	Deep Neural Network
EI	Embedded Intelligence
HMDB	Human Motion Database
ISI	Information Sciences Institute
JPL	Jet Propulsion Laboratory
LoL	Learning with Optimal Labels
LwLL	Learning with Less Labeling
MAP	Maximum A Posteriori
MIT	Massachusetts Institute of Technology
ML	Machine Learning
MLE	Maximum Likelihood Estimate
MNIST	Modified National Institute of Standards and Technology
mAP	Mean Average Precision
NASA	National Aeronautics and Space Administration
NER	Named Entity Recognition
NLP	Natural Language Processing
PAC	Probably Approximately Correct
PCA	Principal Components Analysis
PI	Principal Investigator
STR	Systems & Technology Research
TA 1/2/3	Technical Area 1/2/3
VOC	Visual Object Classes