Conformal Methods for Efficient and Reliable Deep Learning

by

Adam Fisch

B.S.E., Princeton University (2015)
S.M., Massachusetts Institute of Technology (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2023

Authored by: Adam Fisch
Department of Electrical Engineering and Computer Science
August 30, 2023

Certified by: Regina Barzilay
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by: Tommi Jaakkola
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Conformal Methods for Efficient and Reliable Deep Learning

by

Adam Fisch

Submitted to the Department of Electrical Engineering and Computer Science
on August 30, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Deep learning has seen exciting progress over the last decade. As large foundation models continue to evolve and be deployed into real-life applications, an important question to ask is how we can make these expensive, inscrutable models more efficient and reliable. In this thesis, we present a number of fundamental techniques for building and deploying effective deep learning systems that are broadly based on conformal prediction, a model-agnostic and distribution-free uncertainty estimation framework. We develop both theory and practice for leveraging uncertainty estimation to build adaptive models that are cheaper to run, have desirable performance guarantees, and are general enough to work well in many real-world scenarios. Empirically, we primarily focus on natural language processing (NLP) applications, together with substantial extensions to tasks in computer vision, drug discovery, and medicine.

Thesis Supervisor: Regina Barzilay
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Tommi Jaakkola
Title: Professor of Electrical Engineering and Computer Science

Acknowledgements

This work is made possible with the support of numerous individuals that have accompanied me both before and during my graduate school career. I am grateful to all who have contributed to my academic and personal growth throughout this journey.

First and foremost, I would like to extend my heartfelt appreciation to my advisor, Professor Regina Barzilay, for her unwavering guidance, constructive feedback, support, and dedication. Her expertise and mentorship have been invaluable in shaping the direction of my research, and the resulting contributions within this thesis. Regina and the Barzilay lab group have been the foundation of my last six years at MIT.

I am also indebted to my second thesis supervisor and collaborator, Professor Tommi Jaakkola, whose brilliance and infectious enthusiasm have made my graduate school experience both intellectually stimulating and enjoyable. His keen insights and thoughtful suggestions have greatly contributed to the technical depth and quality of my work. Tommi was the first to introduce me to the field of conformal prediction.

I would also like to thank the other members of my committee, Professors Alexander Rush and Amir Globerson, for their time commitment, rigorous examination, and valuable comments that have enriched the quality and clarity of this thesis.

I have been fortunate to be immersed in a vibrant research environment both at MIT and the larger machine learning community, and have had the pleasure of engaging and collaborating with a number of remarkable individuals. I would particularly like to highlight Tal Schuster, a long-time friend and co-author of many of the publications

upon which several of the chapters of this thesis are based. Similarly, I would like to thank a long, yet still incomplete, list of collaborators I have had the great opportunity to work with: Anastasios Angelopoulos, Stephen Bates, Ming-Wei Chang, Danqi Chen, Eunsol Chen, Jon Clark, Tianyu Gao, Jiang Guo, Robin Jia, Bracha Laufer-Goldshtein, Kenton Lee, Lihua Lei, Victor Quach, Minjoon Seo, Dima Smirnov, Alon Talmor, Adam Yala, and others. Yet many more friends and lab-mates have also had a profound influence on my research and development through countless discussions both inside and outside of the lab, both serious and lighthearted.

It is difficult to fully express the depth of my gratitude for the support and companionship of my family and friends, both in Boston and around the country. Friend Group B, Tal Schuster, Dima Smirnov, PJ Gjeltema, Caroline Gelinne, Becki Ho, Max Shatkhin, Craig Minoff, Michelle Scharfstein, Cecillia Lulli, Ankush Gola, Joe and Abby Kelling, Nick Luzzaraga, John Sturm, Margaret Mattes, Chris Moyer, Rachel Bergman, Nathan and Miriam Wasserman have all made my grad school experience so much richer and more enjoyable in its highs, and more bearable in its lows. I am also thankful for my amazing family—Mom, Dad, Mendy, Ben, and cousins. Last but not least, I want to thank my fiancé Zoe for her constant encouragement, acceptance, love, partnership, and use of an eye mask during my late night hours.

Finally, my PhD research has been supported by the NSF Graduate Research Fellowship under Grant #1122374, and largely fueled by copious amounts of Diet Coke.

Thank you all.

Contents

1

Introduction

1.1 Motivation

Steady advancements in deep learning methods in recent years have led to widespread, and at times revolutionary, breakthroughs in fields such as natural language processing (Devlin et al., 2019; Brown et al., 2020; Schulman et al., 2023), computer vision (He et al., 2015; Dosovitskiy et al., 2021), computational chemistry (Jumper et al., 2021), and predictive medicine (Yala et al., 2021; Mikhael et al., 2023). A significant part of this progress can be attributed to *scale*: large foundation models trained on unprecedented amounts of data have changed the way that many predictive tasks are modeled and solved. At the same time, as these models begin to permeate real-life applications, new challenges start to arise. In particular, the large computational footprints of the best modern models makes them expensive to run, and even these best models can—and inevitably do—make harmful mistakes during deployment.

In this thesis, we develop rigorous statistical tools based on conformal prediction that help to tackle multiple aspects of these interrelated challenges. Conformal prediction (Vovk et al., 2005) is an uncertainty estimation framework that has become increasingly popular in the machine learning community due to its favorable model-agnostic, distribution-free, and finite-sample guarantees. We build on conformal prediction to propose several fundamental theoretical and empirical advancements that help prepare users to safely use deployed models in difficult, but common, situations that arise in the real world. At the same time, we also show how these uncertainty estimation techniques can be leveraged to make more efficient predictions by taking the opposite approach: for easy inputs, it can pay to be less conservative, and to choose to use less expensive, simpler functions to make a prediction—but still ensure that any degradation suffered to the model's overall performance is tightly bounded.

Efficient computation in large neural networks

Large, multi-layered neural networks such as Transformers (Vaswani et al., 2017b) have become the de facto standard approach for solving tasks in natural language processing and beyond. Despite their impressive performance, however, their often massive computational burden makes them costly to run. Concerns about their efficiency have kindled a large body of research in the field (Schwartz et al., 2020a). Making models more efficient does not generally come for free: naïve techniques for speeding up inference can result in unpredictable hits to dependent dimensions, such as model accuracy, especially in the worst case over harder, minority sub-groups. A key insight, however, is that this degradation can vary from input to input—and not all examples require the same amount of computation (e.g., simple functions can be used to infer their labels). We develop techniques that allow for adaptive computation in neural networks that depends on the complexity of the input examples, together with precise probabilistic upper bounds on the increases in error that may be suffered.

Rigorous, general-purpose uncertainty estimation

Making models more efficient to run allows them to be more widely deployed in practical scenarios where computational constraints may be limiting. As models are more widely deployed in the real world, however, they are at risk of making costly mistakes. Most modern systems output a single prediction—whether that is a real value, label, free-form generated text, structured object, or other response variable. For many applications, however, it is also critical to enrich such predictions with meaningful uncertainty estimates (Amodei et al., 2016; Jiang et al., 2012, 2018; Rajpurkar et al., 2018). In sensitive high-stakes applications (such as those in medicine), displaying confidence metrics (that actually reflect whether the model is likely to be right or wrong) can be as important as obtaining high-accuracy. Uncertainty estimation is also relevant any time a user is not able to easily verify the answer themselves, and must otherwise trust it blindly. For example, most machine translation system users have no way of knowing if a particular translation is accurate. Reliable uncertainty estimates can mitigate some of the negative consequences of these errors. A model that is aware of its own uncertainty can be used to (1) tell the user how sure it is with some probability, (2) say that it is confident that the right answer is one of a few options, or (3) abstain from predicting altogether in order to defer to a different model or a human. In this thesis, we explore several of these directions, and build on conformal prediction to establish additional important groundwork for calibrating set-valued uncertainty estimates with provable performance guarantees. Specifically, we propose extensions to conformal prediction that provably control various types of risks, are better suited for large label spaces with non-unique answers=, are amenable to being used in few-shot settings with limited calibration data for validation, and are generally more useful when applied to practical problems with constraints.

Applications

This thesis contains both theoretical and empirical contributions. On the applied side, our primary focus is on natural language processing tasks such as automatic fact verification, sentiment analysis, open-domain question answering, and information extraction or retrieval. However, we also note that our methods are general in nature, and easily extend to domains beyond language. As a demonstration, a significant number of our experiments explore tasks in computational chemistry for drug discovery, computer vision, and medicine, with strong results. Code for our experiments is available online (see each chapter for details).

Overview

In the rest of this chapter, we discuss related work, and give background on some of the key results in conformal prediction that we build on here. We then conclude this introduction with an outline of the key ideas of the remaining chapters of this thesis.

1.2 Related work

Efficiency and reliability have been popular topics in the machine learning literature for decades, well before models reached the performance levels that they have today. Our work builds on top of this strong foundation. In the following, we briefly summarize some of past and present developments related to our area of focus. Note that additional related work is also provided within each chapter of this thesis.

Adaptive computation

Improving inference-time efficiency of modern networks such as LLMs has been an ongoing effort of the research community over the past several years (Moosavi et al., 2020; Tay et al., 2022; Xu et al., 2021), leveraging techniques such as knowledge distillation (Bai et al., 2020; Hinton et al., 2015; Jiao et al., 2019; Sun et al., 2019; Wang et al., 2020a; Sanh et al., 2019), floating point quantization (Sun et al., 2020; Shen et al., 2020), layer pruning (Fan et al., 2020a), vector dropping (Kim and Cho, 2021), and others (Lei, 2021). Another line of work involves conditional computation to train larger models that only use a sparse subset of the full network during inference, for example by routing over mixture-of-experts (Bapna et al., 2020; Du et al., 2021; Kudugunta et al., 2021; Zhou et al., 2022), recurring modules (Dehghani et al., 2019; Graves, 2016; Jernite et al., 2016), or accessing external memory (Graves et al., 2014; Sukhbaatar et al., 2015; Wu et al., 2022). These models, however, still use a similar amount of compute for all input examples.

In this thesis, we focus on *adaptive compute*, a specific kind of conditional compute that aims to dynamically allocate different computational power per example, with the goal of reducing the overall complexity while maintaining high performance. This approach, often referred to as *early-exiting* (Cambazoglu et al., 2010; Figurnov et al., 2017; Liu et al., 2022; Teerapittayanon et al., 2016; Wang et al., 2018b; Yin et al., 2021), is complementary to many of the solutions above and can potentially be combined with them. Multiple early-exit techniques for Transformers (e.g., BERT (Devlin et al., 2019)) have been recently proposed (Baier-Reinio and Sterck, 2020; Hou et al., 2020; Li et al., 2020; Liu et al., 2020b, 2021a; Schwartz et al., 2020b; Stickland and Murray, 2019; Xin et al., 2020c; Zhou et al., 2020b; Zhu, 2021). Most of these methods rely on intrinsic confidence measures (e.g., based on the softmax distribution), while others try to predict the routing in advance (Liu et al., 2021b; Sun et al., 2022), or train a small early-exit classifier (Xin et al., 2021), as we also examine here. Importantly, however, we not only focus on adaptation, but also *calibrated* adaptation in which we are able to provably control for the differences in our new predictions.

Uncertainty estimation

A large body of work in estimating model uncertainty focuses on calibrating model-based conditional probabilities, $p_\theta(\hat{y} \mid x)$. Calibration has a rich history in machine learning (Brier, 1950; Murphy and Epstein, 1967; Dawid, 1982; Foster and Vohra, 1998; Gneiting et al., 2007; Niculescu-Mizil and Caruana, 2005; Guo et al., 2017a), and has generally been used to describe the quality where the model's predicted probabilities match that its actually observed accuracy, i.e., $\mathbb{P}(\hat{y} = y \mid p_\theta(\hat{y} \mid x) = \alpha) = \alpha$, though more sophisticated definitions have also been proposed (Vaicenavicius et al., 2019; Zhao et al., 2021; Gupta and Ramdas, 2022). Recently, it has begun to experience a resurgence in the deep learning literature (Kuleshov and Liang, 2015; Kuleshov et al., 2018; Kumar et al., 2019; van Amersfoort et al., 2020; Gupta et al., 2020a), partly motivated by observations that modern neural networks can be significantly miscalibrated out-of-the-box (Guo et al., 2017a; Ashukha et al., 2020). To start, a number of efforts have focused on how to best define and measure calibration, especially in multi-class settings. A common approach (e.g., as taken by Guo et al. (2017a)) measures the top-label calibration error, or the probability that the model's top prediction is correct (a definition we adopt). Other methods propose more precise notions of calibration, including calibration that is conditioned on the predicted class (Gupta and Ramdas, 2022), across classes (Kull et al., 2019), for sub-populations or individuals (Hebert-Johnson et al., 2018; Zhao et al., 2020), or with respect to decision making (Zhao et al., 2021). In theory, estimates of these types could be used to create prediction sets (Gupta et al., 2020b), but they are not always accurate (Guo et al., 2017b; Ashukha et al., 2020; Hirschfeld et al., 2020).

In a similar vein, Bayesian formalisms quantify uncertainty via computing the posterior predictive distribution over model parameters (Neal, 1996; Graves, 2011; Hernández-Lobato and Adams, 2015; Gal and Ghahramani, 2016). Indeed, Bayesian posteriors can be used to create so-called credible intervals, which have analogous interpretations to some of the types of confidence intervals that we develop in this thesis. However, the quality of these methods can vary depending on the suitability of the presumed prior and on approximation error, though some recent work has also combined Bayesian methods with frequentist coverage guarantees (Hoff, 2021). Selective classification (Hellman, 1970; De Stefano et al., 2000; Herbei and Wegkamp, 2006; El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017), where models have the option to abstain from answering when not confident, is also similar in motivation, but still complimentary, to a number of the techniques that we develop here. In fact, selective classification can be considered as a special case of set-based conformal prediction in which the classifier chooses to abstain unless the predicted confidence set is a singleton.

This thesis focuses on branch of uncertainty estimation called conformal prediction, that is based on distribution-free, finite-sample guarantees. In contrast to Bayesian methods, no assumptions are made on the distribution family or prior, and in contrast to typical calibration approaches, the types of guarantees are not asymptotic in the number of data points used. We cover this more in depth next, and in §1.3.

Conformal prediction

Conformal prediction was developed by Vladimir Vovk and collaborators beginning in the late 1990s (Vovk et al., 1999, 2005), and has recently become a popular uncertainty estimation tool in the machine learning community, due to its favorable model-agnostic, distribution-free, finite-sample guarantees. In its standard formulation, conformal prediction uses n calibration points $(X_1, Y_1), i = 1, \dots n$ to construct a prediction set for the $(n+1)$th example, $\mathcal{C}_\epsilon(X_{n+1})$, that satisfies guarantees on the event $\mathbf{1}\{Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})\}$. Recently there have been many extensions of the conformal algorithm, mainly targeting deviations from exchangeability (Tibshirani et al., 2019; Gibbs and Candes, 2021; Barber et al., 2022; Fannjiang et al., 2022) and improved conditional coverage (Barber et al., 2020; Romano et al., 2019a; Guan, 2020; Romano et al., 2020b; Angelopoulos et al., 2021b). Most relevant to us is recent work on risk control in high probability (Vovk, 2012; Bates et al., 2020; Angelopoulos et al., 2021a) and its applications (Park et al., 2020; Sankaranarayanan et al., 2022; Angelopoulos et al., 2022b,c; Cauchois et al., 2021; Stanton et al., 2023, *inter alia*). We provide additional background on conformal prediction in the next section.

1.3 Background

We begin with a brief review of conformal prediction. See also Angelopoulos and Bates (2021) for a modern introduction to the area, or Shafer and Vovk (2008) for a more classical alternative. Here, and in the rest of the chapter, upper-case letters (X) denote random variables; lower-case letters (x) denote scalars, and script letters $(\mathcal{X})$ denote sets, unless otherwise specified. Proofs are deferred to the appendix.

Suppose we have been given n examples, $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, as calibration data, that have been drawn exchangeably from some underlying distribution P. Let $X_{n+1} \in \mathcal{X}$ be a new exchangeable test example for which we would like to make a prediction. The goal of conformal predictin is to use the first n examples to construct a prediction set $\mathcal{C}_\epsilon$ that contains Y_{n+1} at a tolerance level $\epsilon \in (0, 1)$, i.e.,

$$\mathbb{P}\Big(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1}\Big) \geq 1 - \epsilon; \quad \text{for any distribution } P. \tag{1.1}$$

The above probability is marginal over all $n + 1$ data points.

At the core of the conformal algorithm is a simple statistical hypothesis test: for each candidate $y \in \mathcal{Y}$ we must either accept or reject the null hypothesis,[1]

$$H_0 \colon y \text{ is the correct label for } X_{n+1}. \tag{1.2}$$

The test statistic for this hypothesis test is a *nonconformity measure*. Though $\mathcal{N}$ can be any scoring function, traditionally $\mathcal{N}$ compares the proposed point (x, y) to a dataset of exchangeable examples $\mathcal{D}$ to measure how plausible the pairing is. Informally, a lower value of $\mathcal{N}$ reflects that point (x, y) "conforms" to $\mathcal{D}$, whereas a higher value of $\mathcal{N}$ reflects that (x, y) is atypical relative to $\mathcal{D}$. A practical choice for $\mathcal{N}$ is a model-based loss, e.g., $-\log p_\theta(y|x)$, where θ is a model fit to $\mathcal{D}$. An important caveat, however, is that $\mathcal{N}$ should preserve exchangeability (which we will explain later).

Definition 1.3.1 (Nonconformity measure)**.** *Let $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ be the space of examples (X, Y), and let $\mathcal{Z}^{(*)} := \bigcup_{d \geq 1}(\mathcal{X} \times \mathcal{Y})^d$ be the space of datasets of examples $\mathcal{D}$, of any size $d \geq 1$. A nonconformity measure $\mathcal{N}$ is then a measurable mapping $\mathcal{N} \colon \mathcal{Z} \times \mathcal{Z}^{(*)} \to \mathbb{R}$, that assigns a real-valued score to any example (X, Y), indicating how different it is from a reference dataset $\mathcal{D}$.[2] Furthermore, in order to retain exchangeability, $\mathcal{N}$ should be symmetric with respect to permutations of its inputs.*

A key assumption in conformal prediction is that of the exchangeability of its input data points. An exchangeable sequence of n random variables is a sequence whose joint probability distribution is a symmetric function of its n arguments.

[1] For background on hypothesis testing (and multiple hypothesis testing), see Appendix A.2.

[2] The definition of "different" here is intentionally vague, as any metric will technically work.

Definition 1.3.2. *Random variables $X_1, \dots, X_n$ are said to be exchangeable if*

$$(X_1, \dots, X_n) \overset{d}{=} (X_{\pi(1)}, \dots, X_{\pi(n)}) \quad \textit{for any permutation } \pi. \tag{1.3}$$

Exchangeability is important, since it will allow us to make valid probabilistic statements about the new data point, as we will show later. One way to retain exchangeability is to treat calibration and test points symmetrically. For example, let $Z_j := (X_j, Y_j)$, $j = 1, \dots, n$ be the training data. Then, for test point $x \in \mathcal{X}$ and candidate label $y \in \mathcal{Y}$, we calculate the *nonconformity scores* for (x, y) as:

$$V_j^{(x,y)} := \mathcal{N}(Z_j, Z_{1:n} \cup \{(x, y)\}) \text{ and} \tag{1.4}$$

$$V_{n+1}^{(x,y)} := \mathcal{N}((x, y), Z_{1:n} \cup \{(x, y)\}). \tag{1.5}$$

This formulation, known as *full* conformal prediction, requires running the learning algorithm that underlies $\mathcal{N}$ potentially many times for every new test point ($|\mathcal{Y}|$ times). "Split" conformal prediction (Papadopoulos, 2008) uses a held-out training set to first learn $\mathcal{N}$, and then fix it. The nonconformity scores are then just computed as

$$V_j := \mathcal{N}(Z_j) \text{ and } V_{n+1}^{(x,y)} := \mathcal{N}((x, y)), \tag{1.6}$$

which preserves exchangeability of $(V_1, \dots, V_{n+1}) = (\mathcal{N}(X_1, Y_1), \dots, \mathcal{N}(X_{n+1}, Y_{n+1}))$. This is a more computationally attractive alternative, but comes at the expense of predictive efficiency when data is limited (since we need to reserve separate data for learning $\mathcal{N}$). As we are primarily dealing with expensive neural models, we adopt this approach in most of this work.[3] Exchangeability then allows us to quantify how nonconformal a new point is using a rank-based p-value.

[3] Split conformal prediction also allows for simple nonconformity score calculations for regression tasks. For example, assume that a training set has been used to train a fixed regression model, $f_\theta(x)$. The absolute error nonconformity measure, $|y - f_\theta(x)|$, can then be easily evaluated for all $y \in \mathbb{R}$. Furthermore, since the absolute error monotonically increases going away from $f_\theta(x)$, the conformal prediction $\mathcal{C}_\epsilon$ simplifies to a closed-form *interval*.

Lemma 1.3.3 (Rank-based p-value). *Assume that the random variables* $V_1, \ldots, V_{n+1}$ *are exchangeable. We define the smoothed rank-based p-value* $\texttt{pval}(V_{n+1}, V_{1:n})$ *as*

$$\texttt{pval}(V_{n+1}, V_{1:n}) := \frac{|\{i \in [1,n] \colon V_i > V_{n+1}\}| + \tau|\{i \in [1,n] \colon V_i = V_{n+1}\}| + 1}{n+1}, \quad (1.7)$$

where $\tau \sim U(0,1)$*. Then, for any* $\epsilon \in (0,1)$*, we have* $\mathbb{P}(\, \texttt{pval}(V_{n+1}, V_{1:n}) \leq \epsilon \,) \leq \epsilon$*.*

Proof. See Appendix A.1.1. □

To construct the final conformal *prediction*, the classifier uses the p-values to include all y for which the null hypothesis—i.e., that the candidate pair (x_{n+1}, y) is *conformal*—is not rejected, simply by comparing the computed p-values to ϵ.

Theorem 1.3.4 (Split CP; Vovk et al. (2005); Papadopoulos (2008)). *Assume that the random variables* $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$*,* $i = 1, \ldots, n+1$ *are exchangeable. For a fixed nonconformity measure* $\mathcal{N}$*, let random variable* $V_i = \mathcal{N}(X_i, Y_i)$ *be the score of* (X_i, Y_i)*. For* $\epsilon \in (0,1)$*, define the prediction (based on the first* n *examples) at* $x \in \mathcal{X}$ *as*

$$\mathcal{C}_\epsilon(x) := \Big\{ y \in \mathcal{Y} \colon \texttt{pval}\Big(\mathcal{N}(x,y),\, V_{1:n}\Big) > \epsilon \Big\}. \quad (1.8)$$

Then $\mathcal{C}_\epsilon(X_{n+1})$ *satisfies* $\mathbb{P}\left(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})\right) \geq 1 - \epsilon$*.*

Proof. See Appendix A.1.2. □

This can also be reformulated in terms of quantiles of the distribution of the calibration data, where for a random variable V with distribution function F we define

$$\text{Quantile}(\beta; F) := \inf\{v \colon F(v) \geq \beta\}. \quad (1.9)$$

This quantile can then be computed over the (inflated) calibration set and used directly as a threshold for rejecting $y \in \mathcal{Y}$ based on its non-conformity score.

Corollary 1.3.5. *Under the same conditions as Theorem 1.3.4, define the prediction (based on the first n examples) at $x \in \mathcal{X}$ as*

$$\mathcal{C}_\epsilon(x) := \Big\{y \in \mathcal{Y} \colon \mathcal{N}(x, y) \leq \text{Quantile}(1 - \epsilon;\, V_{1:n} \cup \{\infty\})\Big\} \tag{1.10}$$

Then $\mathcal{C}_\epsilon(X_{n+1})$ satisfies $\mathbb{P}(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})) \geq 1 - \epsilon$.

Proof. See Appendix A.1.3. □

While the above methods give remarkable theoretical guarantees, naïvely applying CP in practice can still yield disappointing results in practice. For example, though these results hold (marginally) for any n, in practice n must be fairly large (e.g., 1000) to achieve reasonable, stable performance—in the sense that $\mathcal{C}_\epsilon$ will not be too large on average (Lei et al., 2018; Bates et al., 2020). In the remainder of this thesis we will both leverage and adapt conformal prediction for various practical use-cases.

1.4 Outline

The rest of this thesis is organized as follows:

- **Chapter 2** presents our first contribution with respect to confident adaptive computation, where we pair early exiting techniques for more efficient inference with performance guarantees via an extension of conformal prediction. Here our focus is not on absolute performance, but rather *relative* performance. Our goal is to make models faster while still retaining some amount of fidelity towards the original predictions (without any early exiting). This work is based on a previously published paper at EMNLP 2021 (Schuster et al., 2021b). Follow-up work (that is not included in this thesis) can also be found in Schuster et al. (2022), where we extend these ideas to adaptive prediction for language modeling.

- **Chapter 3** transitions from more classical classification and regression tasks with well-defined targets to more "messy" open-ended problems with many possible responses, multiple of which are acceptable. This chapter introduces a relaxation to the conformal prediction coverage criterion to handle these types of problems, in addition to a valid way of chaining models of varying complexity together in order to more quickly explore candidates within these large output spaces. An example of this is extractive open-domain question answering, where models both retrieve and read documents at scale (for example, over all of Wikipedia). By relaxing our correctness criterion, and chaining together more efficient models (like BM25 keyword search) with more expensive models (like deep neural networks), we can make predictions that have meaningful guarantees while remaining tractable. This work is based on a previously published paper at ICLR 2021 (Fisch et al., 2021b).

- **Chapter 4** continues to develop theory and practice for problems for which coverage guarantees of the form $\mathbb{P}(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})) \geq 1 - \epsilon$ may take a backseat to other constraints, such as how many false positives within $\mathcal{C}_\epsilon(X_{n+1})$ are tolerable. Besides the different style of guarantee, the main idea advanced in this chapter is to view conformal prediction as a holistic set prediction problem, rather than only considering candidates for $\mathcal{C}_\epsilon(X_{n+1})$ individually. Instead, we develop an auxiliary mechanism for searching and scoring complete candidate sets. This work is based on a previously published paper at ICML 2022 (Fisch et al., 2022).

- **Chapter 5** looks at the problem of data availability, and tries to alleviate some of conformal prediction's heavy reliance on both substantial amounts of training and calibration data in order to obtain effective predictions with guarantees. We do this by leveraging auxiliary data within a meta learning framework. The key idea is to take advantage of additional exchangeability assumptions (this time over the collection of prediction tasks we are interested in solving), to help improve the quality of the (exchangeable) test task's uncertainty estimates. This work is based on a previously published paper at ICML 2021 (Fisch et al., 2021c).

- **Chapter 6** takes a step back to present a powerful theoretical generalization of conformal prediction that allows for valid control of the expectation of not only coverage losses of the form $\mathbf{1}\{Y \in \mathcal{C}_\epsilon(X)\}$, but also of any monotone loss function. In particular, the framework established in this chapter generalizes simple versions of each of the previous chapters, though the analysis in the preceding chapters is different and of independent interest (beyond the empirical components). This work is based on a previous arXiv pre-print (Angelopoulos et al., 2022a).

- **Chapter 7** summarizes this thesis, and highlights directions for future work.

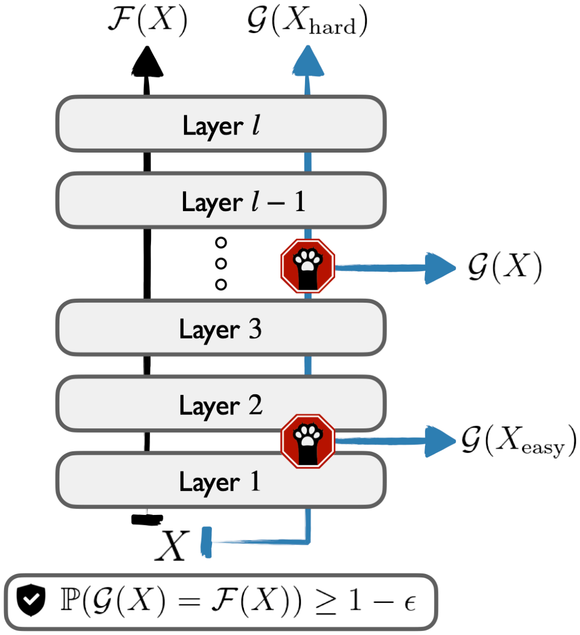
2

Confident Adaptive Transformers

Large and expensive multilayer neural networks are now ubiquitous in machine learning—and natural language processing (NLP) in particular in the form of pre-trained Transformers (Vaswani et al., 2017a). In this chapter we develop an approach for reducing their inference cost while also controlling for predictive performance. Amortized or approximate computational methods increase efficiency, but can come with unpredictable degradations in model quality. We present a method called CATs (**C**onfident **A**daptive **T**ransformers), in which we simultaneously increase computational efficiency, while *guaranteeing* a specifiable degree of consistency with the original model with high confidence. Our method trains additional prediction heads on top of intermediate layers, and dynamically decides when to stop allocating computational effort to each input using a meta consistency classifier. To calibrate our early prediction stopping rule, we formulate a unique extension of conformal prediction. We demonstrate the effectiveness of this approach on classification and regression tasks.

2.1 Introduction

Large pre-trained language models have become the de facto standard approach for solving natural language processing tasks (Devlin et al., 2019; Liu et al., 2019). Despite their impressive performance, however, their often massive computational bur-

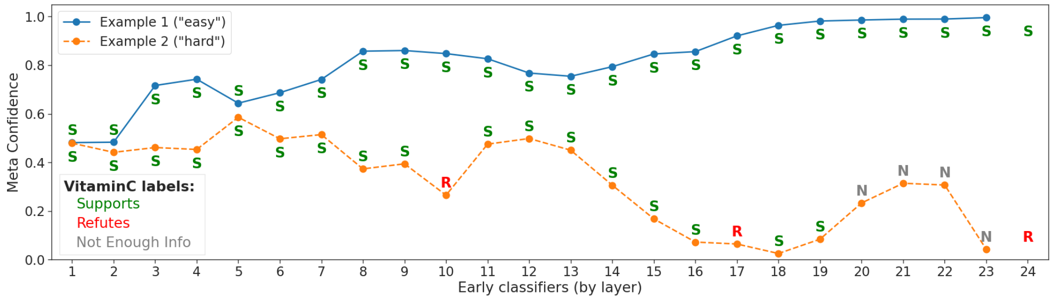


$\mathcal{F}(X)$
$\mathcal{G}(X_{\text{hard}})$
Layer l
Layer $l-1$
$\mathcal{G}(X)$
Layer 3
Layer 2
$\mathcal{G}(X_{\text{easy}})$
Layer 1
X
$\mathbb{P}(\mathcal{G}(X) = \mathcal{F}(X)) \geq 1 - \epsilon$

Figure 2-1: Our CAT model $\mathcal{G}$ can save computational resources by exiting early on certain inputs—while guaranteeing predictive consistency with the full model $\mathcal{F}$.

den makes them costly to run (Schwartz et al., 2020a; Sharir et al., 2020). Concerns about their efficiency have kindled a large body of research in the field (Sanh et al., 2020; Schwartz et al., 2020b; Fan et al., 2020a). For multilayered architectures such as the Transformer, a popular approach is *adaptive early exiting* (Schwartz et al., 2020b; Xin et al., 2020a, *inter alia*). Early exiting takes advantage of the observation that task instances vary in complexity. In this setting, "early" classifiers are added on top of the simpler features of intermediate layers in the base model, and can trigger a prediction before the full model is executed. Naively deciding when to preempt computation, however, can result in unpredictable decreases in model accuracy.

Quantifying the *uncertainty* in a prediction in order to decide when additional computation is needed (or not) is critical to making predictions quickly without excessively sacrificing performance. In this chapter, we present **C**onfident **A**daptive **T**ransformers (CATs), a method for increasing Transformer-based model efficiency while remaining *confident* in the quality of our predictions. Specifically, given a fixed, expensive l-layer model $\mathcal{F}(x)$, we create an amortized model $\mathcal{G}(x)$ that includes early classifiers



(Ex.1) Claim: All airports in Guyana were closed for all international passenger flights until 1 May 2020.
Evidence: Airports in Guyana are closed to all international passenger flights until 1 May 2020.

(Ex.2) Claim: Deng Chao broke sales record for a romantic drama.
Evidence: The film was a success and broke box office sales record for mainland-produced romance films.

Figure 2-2: Confidence levels given by our meta model regarding the *consistency* of our prediction as computation progresses. Ex.1 from the VitaminC fact verification dataset is "easy", and is classified consistently by all early classifiers $\mathcal{F}_k$ (`Supports`). The meta confidence captures this, and increases with time. Ex.2 is harder—and the prediction changes (`Refutes`/`NEI`) as it propagates though the Transformer layers. Appropriately, the meta confidence is low. The exact exit layer of $\mathcal{G}$ is determined as a function of a user-specified tolerance ϵ, see Eq. (2.1).

$\{\mathcal{F}_1, \ldots, \mathcal{F}_l\}$.[1] We then make $\mathcal{G}$ provably consistent with the original $\mathcal{F}$ with arbitrarily high probability (e.g., 95% of the time). See Figure 2-1.

Our approach builds on conformal prediction (CP). As introduced in Chapter 1, CP is a framework for creating well-calibrated predictions (Vovk et al., 2005). Concretely, suppose we have been given n *unlabeled* examples, $X_i \in \mathcal{X}$, $i = 1, \ldots, n$, as calibration data, that have been drawn exchangeably from some underlying distribution P. Let $X_{n+1} \in \mathcal{X}$ be a new exchangeable test example for which we would like to make a prediction. The aim of our method is to construct $\mathcal{G}$ such that its predictions probabilistically agree with those of $\mathcal{F}$ at a tolerance level $\epsilon \in (0, 1)$, i.e.,

$$\mathbb{P}\Big(\mathcal{G}(X_{n+1}) = \mathcal{F}(X_{n+1})\Big) \geq 1 - \epsilon. \tag{2.1}$$

[1] We simply define the final $\mathcal{F}_l$ as $\mathcal{F}_l(x) \triangleq \mathcal{F}(x) \ \ \forall x$.

We consider $\mathcal{G}$ to be ϵ-*consistent* if the frequency of error, $\mathcal{G}(X_{n+1}) \neq \mathcal{F}(X_{n+1})$, does not exceed ϵ.[2] By design, this ensures that $\mathcal{G}$ preserves at least $(1-\epsilon)$-fraction of $\mathcal{F}$'s original performance. Within these constraints, the remaining challenge is to make $\mathcal{G}$ relatively efficient (for example, a consistent, but vacuous, model is simply the identity $\mathcal{G} \triangleq \mathcal{F}$, which does not save any computation time).

In order to support an efficient $\mathcal{G}$, we need a reliable signal for inferring whether or not the current prediction is likely to be stable. Past work (e.g., Schwartz et al., 2020b) rely on potentially poorly calibrated metrics such as the early classifier's softmax response. We address this challenge by instead directly learning meta "consistency predictors" for each of the $l-1$ early classifiers of our l layer model, by leveraging patterns in past predictions.[3] Figure 2-2 demonstrates the progression of meta confidence scores across layers when applied to "easy" versus "hard" instances from the VitaminC fact verification task (Schuster et al., 2021a).

We pair the scores of our meta classifier for each layer with a stopping rule that is calibrated using a unique twist on standard conformal prediction. Traditionally, CP is used to construct prediction *sets* that cover the desired target (e.g., Y_{n+1}) with high probability. We invert the CP problem to first infer the multi-label set of *inconsistent* layers, and then exit at the first layer that falls in its complement. We then demonstrate that this can be reduced to setting a simple (but well-calibrated) exit threshold for the meta classifier scores. Our resulting algorithm is (1) fast to compute in parallel to the main Transformer, (2) requires only unlabeled data, and (3) is statistically efficient in practice, in the sense that it finds low exit layers on average while still maintaining the required predictive consistency.

We validate our method on four diverse NLP tasks—covering both classification and regression, different label space sizes, and varying amounts of training data. We find that it constitutes a simple-yet-effective approach to confident adaptive prediction with minimal interventions and desirable theoretical guarantees.

[2] For regression, we define equality as $|\mathcal{G}(\cdot) - \mathcal{F}(\cdot)| \leq \tau$, for some specified τ.

[3] We refer to the meta aspect of the classifier (not to be confused with *meta-learning*).

Contributions. In summary, the main results of this chapter are as follows:

1. An extension of conformal prediction to accommodate adaptive prediction;
2. A meta consistency classifier for deriving a confident "early exiting" model;
3. A demonstration of the utility of our framework on both classification and regression tasks, where we show significant efficiency gains with high consistency.

2.2 Related work

Adaptive computation. Reducing the computational cost of neural models has received intense interest. Adaptive approaches adjust the amount of computation per example to *amortize* the total inference cost (see Teerapittayanon et al., 2017; **?**; Huang et al., 2018; Kaya et al., 2019; Wang et al., 2018a, *inter alia*). As discussed in §2.1, our method is inspired by the approach of Schwartz et al. (2020b) and others (Liu et al., 2020a; Geng et al., 2021; Zhou et al., 2020a), where they preempt computation if the softmax value of any early classifier is above a predefined threshold. Yet unlike our approach, their model is not guaranteed to be accurate. In concurrent work, Xin et al. (2021) propose a meta confidence classifier similar to ours. However, as in previous work, they do not do any calibration that guarantees consistency.

Conformal prediction. CP (Vovk et al., 2005) typically is formulated in terms of prediction sets $\mathcal{C}_\epsilon(X_{n+1})$ with finite-sample, distribution-free guarantees on the event that $\mathcal{C}_\epsilon$ contains Y_{n+1} (see §1.3). As we discuss in §2.4, internally our method follows a similar approach in which we try to conservatively identify the inadmissible set of all layers that are *inconsistent* (and exit at the first layer that falls in that set's complement). Most relevant to our work, Cauchois et al. (2021) presents algorithms for conformal multi-label predictions. We leverage similar methods in our model, but formulate our solution in terms of the *complement* of a multi-label set of inconsistent *predictions*. Our work adds to recent directions that explore CP in the context of risk-mitigating applications (Lei and Candès, 2020; Romano et al., 2020a; Bates et al., 2020; Fisch et al., 2021a), or meta-learning settings (Fisch et al., 2021d).

2.3 Early exiting transformers

In the following, we describe our dynamic early exiting model. We summarize early classification (following previous work) for convenience (§2.3.1), and then present our proposed meta consistency classifier (§2.3.2). We focus on classification and regression tasks, given a model $\mathcal{F}(x) = y$. We assume that $\mathcal{F}$ maps the input $x \in \mathcal{X}$ into a series of feature representations before making the prediction $y \in \mathcal{Y}$. Here, $\mathcal{F}$ is a multilayered Transformer (Vaswani et al., 2017a) composed of l layers (although our method can be applied to any multilayer network).

For all of our downstream tasks we follow standard practice and assume that the input contains a `[CLS]` token whose hidden representation is used for prediction. For classification, we use a task-specific head, $\text{softmax}(\mathbf{W}_o(\phi(\mathbf{W}_p\mathbf{h}_{\texttt{[CLS]}})))$, where $\mathbf{h}_{\texttt{[CLS]}} \in \mathbb{R}^d$ is the hidden representation of the `[CLS]` token,[4] ϕ is a nonlinear activation, and $\mathbf{W}_*$ are linear projections, where $\mathbf{W}_p \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{Y}|\times d}$. Regression is treated similarly, but uses a 1-d output projection instead of a softmax, $\mathbf{w}_o \cdot \mathbf{h}_{\texttt{[CLS]}}$.

2.3.1 Early predictors

$\mathcal{F}$'s structure yields a sequence of hidden `[CLS]` representations, $\{\mathbf{h}^{(1)}_{\texttt{[CLS]}}, \ldots, \mathbf{h}^{(l)}_{\texttt{[CLS]}}\}$, where $\mathbf{h}^{(k)}_{\texttt{[CLS]}} \in \mathbb{R}^d$ is the representation after applying layer k. After each intermediate layer $k < l$, we train an early classification head that is similar to the head used in $\mathcal{F}$, but reduce the dimensionality of the first projection to $\mathbf{W}^{(k)}_p \in \mathbb{R}^{d_e\times d}$.[5] The final $\mathcal{F}_l$ is unchanged from $\mathcal{F}$. These extra $(l-1) \times (d_e \times d + d_e \times |\mathcal{Y}|)$ parameters are quick to tune on top of a fixed $\mathcal{F}$, and we can reuse $\mathcal{F}$'s training data as $\mathcal{D}_{\text{tune}}$.[6] The classifier $\mathcal{F}_k(x) = \text{softmax}(\mathbf{W}^{(k)}_o(\phi(\mathbf{W}^{(k)}_p\mathbf{h}^{(k)}_{\texttt{[CLS]}})))$ is then used after layer k to get an early prediction candidate. Early regression is handled similarly.

[4] d varies by $\mathcal{F}$. In Albert-xlarge $d = 2048$.

[5] This is purely for efficiency. We set $d_e = 32$ in all our experiments.

[6] Or if $\mathcal{D}_{\text{tune}}$ is unlabeled, we can use $\mathcal{F}(x)$ as labels.

Meta Feature	Description
$\hat{y}_k$	The current prediction.
history	The past $k-1$ predictions, $\hat{y}_{1:k-1}$ (for classification we use the confidence in the current prediction $p_i(\hat{y}_k \mid x)$, for $i \in [k-1]$).
p_k^{max}	Probability of the prediction, $p_k(\hat{y}_k \mid x)$.
p_k^{diff}	Difference in probability of the top predictions, $p_k(\hat{y}_k \mid x) - \text{argmax}_{y_k \neq \hat{y}_k}\ p_k(y_k \mid x)$.

Table 2.1: Additional meta features used as input to the meta early exit classifier, $\mathcal{M}_k$. Where specified, the probability p_k is taken from the model's early classifier softmax. The features p_k^{max} and p_k^{diff} are only used for classification tasks.

2.3.2 Meta early exit classifier

To decide *when* to accept the current prediction and stop computation, we require some signal as to how likely it is that $\mathcal{F}_k(x) = \mathcal{F}(x)$. Previous work relies on intrinsic, and mostly heuristic, measures like the classifier's softmax response. Here, we present a *meta* classifier to explicitly estimate the consistency of an early predictor. Given fixed $\mathcal{F}_k$ and $\mathcal{F}$, we train a small binary MLP, $\mathcal{M}_k(x) \in \mathbb{R}$, on another *unlabeled* (but limited) sample of task in-domain data, $\mathcal{D}_{\text{meta}}$. As input, we provide the current "early" hidden state $\phi(\mathbf{W}_p^{(k)} \mathbf{h}_{\texttt{[CLS]}}^{(k)})$, in addition to several processed meta features, see Table 2.1. We then train $\mathcal{M}_k$ with a binary cross entropy objective, where we maximize the likelihood of predicting $\mathbf{1}\{\mathcal{F}_k(x_i) = \mathcal{F}(x_i)\}$ for $x_i \in \mathcal{D}_{\text{meta}}$.

Using the trained $\mathcal{F}_k$ and $\mathcal{M}_k$, we define the full adaptive model $\mathcal{G}$ using the rule

$$\mathcal{G}(x; \boldsymbol{\tau}) := \begin{cases} \mathcal{F}_1(x) & \text{if } \mathcal{M}_1(x) > \tau_1, \\ \mathcal{F}_2(x) & \text{else if } \mathcal{M}_2(x) > \tau_2, \\ & \vdots \\ \mathcal{F}_l(x) & \text{otherwise,} \end{cases} \tag{2.2}$$

where $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{l-1})$ are confidence thresholds. The key challenge is to *calibrate* τ_k such that $\mathcal{G}$ guarantees ϵ-consistent performance per Eq. (2.1).

2.3.3 Warmup: development set calibration

A simple approach to setting $\boldsymbol{\tau}$ is to optimize performance on a development set $\mathcal{D}_{\text{dev}}$, subject to a constraint on the empirical inconsistency:

$$\begin{aligned} \boldsymbol{\tau}^* := \underset{(\tau_1,\ldots,\tau_{l-1})}{\text{minimize}} \ & \widehat{\mathbb{E}}_{\text{dev}}[\text{exit}(G(X;\boldsymbol{\tau}))] \\ \text{s.t.} \ \ & \widehat{\mathbb{E}}_{\text{dev}}[\mathbf{1}\{\mathcal{G}(X;\boldsymbol{\tau}) = \mathcal{F}(X)\}] \geq 1-\epsilon, \end{aligned} \tag{2.3}$$

where exit$(\cdot)$ measures the exit layer, and $\widehat{\mathbb{E}}_{\text{dev}}$ is the average over $\mathcal{D}_{\text{dev}}$. Using a standard error bound that leverages the concentration properties of the binomial distribution (Langford, 2005) on another split, $\mathcal{D}_{\text{cal}}$, we can derive the following:

Proposition 2.3.1. *Let X_i, $i = 1, \ldots, n$ be an i.i.d. sample with $s = \sum_{i=1}^n \mathbf{1}\{\mathcal{G}(X_i;\boldsymbol{\tau}) = \mathcal{F}(X_i)\}$. Then, up to a confidence level δ, we have that*

$$\mathbb{P}(\mathbb{P}(\mathcal{G}(X;\boldsymbol{\tau}) = \mathcal{F}(X) \mid \mathcal{D}_{\text{cal}}) \geq 1-\tilde{\epsilon}) \geq 1-\delta, \tag{2.4}$$

where the outer probability is over the draw of $\mathcal{D}_{\text{cal}}$. $\tilde{\epsilon}$ is the solution to $\text{Beta}(s, n-s+1) = \delta$, *where* Beta *is the incomplete beta function.*

Proof. See Appendix B.1.1. □

Though in practice $\tilde{\epsilon}$ might be close to ϵ for most well-behaved distributions and sufficiently sized $\mathcal{D}_{\text{dev}}$ and $\mathcal{D}_{\text{cal}}$, unfortunately Eq. (2.4) does not give a fully *specifiable* guarantee as per Eq. (2.1). Readjusting $\boldsymbol{\tau}$ based on $\mathcal{D}_{\text{cal}}$ requires correcting for multiple testing in order to remain theoretically valid, which can quickly become statistically inefficient. In the next section, we provide a novel calibration approach that allows us to guarantee a target performance level with strong statistical efficiency.

2.4 Conformalized early exits

We now formulate the main contribution of this paper, which is a *distribution-free* and *model-agnostic* method based on CP for guaranteeing any performance bound an end-user chooses to specify. Our training (§2.3), conformal calibration (§2.4), and inference pipelines are summarized in Algorithm 1.

2.4.1 Formulation

Let $\mathcal{I}(x) := \{i \colon \mathcal{F}_i(x) \neq \mathcal{F}(x)\}$ be the index set of layers that are *inconsistent* with the final model's prediction. To maintain ϵ-consistency, we must avoid using any of the predictions specified by this set, $\mathcal{F}_i(x)$ where $i \in \mathcal{I}(x)$, more than ϵ-fraction of the time for $x \in \mathcal{X}$. In §2.4.2, we show how $\mathcal{M}_{1:l-1}$ can be paired with a conformal procedure to obtain calibrated thresholds $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{l-1})$ such that we obtain a conservative prediction of the layer index set $\mathcal{I}(x)$,

$$\mathcal{C}_\epsilon(x) := \{k \colon \mathcal{M}_k(x) \leq \tau_k\}, \tag{2.5}$$

where we ensure that $\mathcal{I}(x) \subseteq \mathcal{C}_\epsilon(x)$ with probability at least $1 - \epsilon$. Proposition 2.4.1 states our guarantee when $\boldsymbol{\tau}$ is paired with $\mathcal{G}$ following Eq. (2.2).

Proposition 2.4.1. *Assume that examples X_i, $i = 1, \ldots, n+1$ are exchangeable. For any $\epsilon \in (0, 1)$, let the index set $\mathcal{C}_\epsilon$ (based on the first n examples) be the output of conformal procedure satisfying*

$$\mathbb{P}(\mathcal{I}(X_{n+1}) \subseteq \mathcal{C}_\epsilon(X_{n+1})) \geq 1 - \epsilon. \tag{2.6}$$

Define $K := \min\{j : j \notin \mathcal{C}_\epsilon(X_{n+1})\}$ to be the index of the layer selected by $\mathcal{G}$. Then

$$\mathbb{P}(\mathcal{F}_K(X_{n+1}) = \mathcal{F}(X_{n+1})) \geq 1 - \epsilon. \tag{2.7}$$

Proof. See Appendix B.1.2. □

Remark 2.4.2. Note that Eq. (2.6) is stricter than necessary. Fundamentally, we only require that $\mathbb{P}(K \notin \mathcal{I}(X_{n+1})) \geq 1 - \epsilon$. Nevertheless, Eq. (2.6) is easier to calibrate, and leads to strong empirical results despite being theoretically conservative.

Remark 2.4.3. During inference we do not fully construct $\mathcal{C}_\epsilon$; it is only used to calibrate $\boldsymbol{\tau}$ beforehand. Instead we exit at the first layer that is not put in $\mathcal{C}_\epsilon$.

2.4.2 Conformal calibration

We now describe our conformal procedures for calibrating $\boldsymbol{\tau}$. Conformal prediction is based on hypothesis testing, where for a given input x and possible output y, a statistical test is performed to accept or reject the null hypothesis that the pairing (x, y) is correct. In our setting, we consider the null hypothesis that layer k is *inconsistent*, and we use $\mathcal{M}_k(x)$ as our test statistic. Since $\mathcal{M}_k$ is trained to predict $\mathbf{1}\{\mathcal{F}_k(x_i) = \mathcal{F}(x_i)\}$, a high value of $\mathcal{M}_k(x)$ indicates how "surprised" we would be if layer k was in fact inconsistent with layer l for input x. Informally, a low level of surprise indicates that the current input "conforms" to past data. To rigorously quantify the degree of conformity via the threshold τ_k for predictor $\mathcal{M}_k$, we use a held-out set of n unlabeled, exchangeable examples, $\mathcal{D}_{\text{cal}}$.

Independent calibration

As a first approach, we construct $\mathcal{C}_\epsilon(x)$ by composing $l - 1$ separate tests for $\mathcal{F}_k(x) \neq \mathcal{F}(x)$, each with significance α_k, where α_k are corrected for multiple testing. Let $v_k^{(1:n,\infty)}$ denote the inflated empirical distribution of inconsistent layer scores,[7]

$$v_k^{(1:n,\infty)} := \Big\{\mathcal{M}_k(x_i) \colon x_i \in \mathcal{D}_{\text{cal}}, \mathcal{F}_k(x_i) \neq \mathcal{F}(x_i)\Big\} \cup \Big\{\infty\Big\}. \tag{2.8}$$

We then define

$$\tau_k^{\text{ind}} := \text{Quantile}\Big(1 - \alpha_k, v_k^{(1:n,\infty)}\Big), \tag{2.9}$$

[7] "Inflating" the empirical distribution is critical to our finite sample guarantee, see Appendix **??**.

and predict the inconsistent index set at $x \in \mathcal{X}$ as

$$\mathcal{C}_\epsilon^{\text{ind}}(x) := \Big\{ k \colon \mathcal{M}_k(x) \leq \tau_k^{\text{ind}} \Big\}. \tag{2.10}$$

The following theorem states how to set each α_k such that τ_k^{ind} yield a valid $\mathcal{C}_\epsilon^{\text{ind}}$.

Theorem 2.4.4. *Let $\alpha_k = \omega_k \cdot \epsilon$, where ω_k is a weighted Bonferroni correction, i.e., $\sum_{k=1}^{l-1} \omega_k = 1$. Then $\mathcal{C}_\epsilon^{\text{ind}}(X_{n+1})$ satisfies Eq. (2.6).*

Proof. See Appendix B.1.3. □

Remark 2.4.5. $\omega_{1:l-1}$ can be treated as hyper-parameters and tuned on a development set $\mathcal{D}_{\text{dev}}$, as long as $\mathcal{D}_{\text{dev}}$ is kept distinct from $\mathcal{D}_{\text{cal}}$.

Shared calibration

$\mathcal{C}_\epsilon^{\text{ind}}$ has the advantage of calibrating each layer independently. As l grows, however, α_k will tend to 0 in order to retain validity (as specified by Theorem 2.4.4). As a result, $\mathcal{C}_\epsilon^{\text{ind}}$ will lose statistical efficiency. Following a similar approach to Cauchois et al. (2021) and Fisch et al. (2021a), we compute a new test statistic, $\mathcal{M}_{\max}$, as

$$\mathcal{M}_{\max}(x) := \max_{k \in [l-1]} \Big\{ \mathcal{M}_k(x) : \mathcal{F}_k(x) \neq \mathcal{F}(x) \Big\}. \tag{2.11}$$

We discard ill-defined values when $\mathcal{M}_{\max}(x) = \max \varnothing$. $\mathcal{M}_{\max}(x)$ reflects the *worst-case* confidence across inconsistent layers for input x (i.e., where $\mathcal{M}_k(x)$ predicts a high consistency likelihood for layer k when layer k is, in fact, *inconsistent*). This worst-case statistic allows us to keep a constant significance level ϵ, even as l grows. Let $m^{(1:n,\infty)}$ denote the inflated empirical distribution,

$$m^{(1:n,\infty)} := \Big\{ \mathcal{M}_{\max}(x_i) \colon x_i \in \mathcal{D}_{\text{cal}}, \exists k \; \mathcal{F}_k(x_i) \neq \mathcal{F}(x_i) \Big\} \cup \Big\{ \infty \Big\}. \tag{2.12}$$

Algorithm 1 Consistent accelerated inference.

Definitions: $\mathcal{F}$ is a multilayered classifier trained on $\mathcal{D}_{\text{train}}$. $\mathcal{D}_{\text{tune}}$, $\mathcal{D}_{\text{meta}}$ and $\mathcal{D}_{\text{scale}}$ are collections of in-domain unlabeled data points (in practice, we reuse $\mathcal{D}_{\text{train}}$ and divide it to 70/20/10%, respectively). $\mathcal{D}_{\text{cal}}$ has in-domain unlabeled examples not in $\mathcal{D}_{\text{train}}$ (in practice, we take a subset of the task's validation set). ϵ is the user-specified consistency tolerance.

```
function Train($\mathcal{F}$, $\mathcal{D}_{\text{tune}}$, $\mathcal{D}_{\text{meta}}$)
    # Learns $\mathcal{F}_{1\ldots l-1}$ and $\mathcal{M}_{1\ldots l-1}$ components
    # of amortized model $\mathcal{G}$ for Eq. (2.2) (see §2.3.1 and §2.3.2).
    Initialize $\mathcal{G}$ from $\mathcal{F}$ and add early prediction heads.
    # (All of $\mathcal{F}$'s base parameters in $\mathcal{G}$ are frozen.)
    Train prediction heads $\mathcal{F}_{1\ldots l-1}$ on $\mathcal{D}_{\text{tune}}$.
    Add meta early exit classifiers $\mathcal{M}_{1\ldots l-1}$ to $\mathcal{G}$.
    # (All of $\mathcal{G}$'s other parameters are frozen.)
    Train meta early exit classifiers $\mathcal{M}_{1\ldots l-1}$ on $\mathcal{D}_{\text{meta}}$.
    Optionally apply temperature scaling using $\mathcal{D}_{\text{scale}}$.
    return $\mathcal{G}$

function Calibrate($\mathcal{G}$, $\mathcal{D}_{\text{cal}}$, $\epsilon$)
    # Sets thresholds $\boldsymbol{\tau}$ of amortized model $\mathcal{G}$ for Eq. (2.2)
    # using shared calibration (see §2.4.2).
    $M \leftarrow \{\infty\}$
    for $x \in \mathcal{D}_{\text{cal}}$ do
        $S \leftarrow \{\}$
        # Record all inconsistent layers for input $x$.
        # Keep the highest (false) confidence score.
        for $k \in [1, l-1]$ do
            if $\mathcal{F}_k(x) \neq \mathcal{F}(x)$ then
                $S \leftarrow S \cup \mathcal{M}_k(x)$
        $M \leftarrow M \cup \max(S)$
    # Share one threshold across layers.
    $\tau^{\text{share}} \leftarrow \text{Quantile}\big(1-\epsilon, M\big)$
    return $[\tau^{\text{share}}] \times (l-1)$

function Predict($\mathcal{G}$, $\boldsymbol{\tau}$, $x$)
    # Implements Eq. (2.2) to exit early with confidence.
    for $k \in [1, l-1]$ do
        Compute the $k$-th prediction head of $\mathcal{G}$, $\mathcal{F}_k(x)$.
        if $\mathcal{M}_k(x) > \tau_k$ then
            return $\mathcal{F}_k(x)$
    # Fallback to prediction using full computation.
    return $\mathcal{F}_l(x)$
```

We then define a single threshold shared across layers,

$$\tau^{\text{share}} = \text{Quantile}\Big(1-\epsilon, m^{(1:n,\infty)}\Big), \tag{2.13}$$

and predict the inconsistent index set at $x \in \mathcal{X}$ as

$$\mathcal{C}_{\epsilon}^{\text{share}}(x) = \Big\{k \colon \mathcal{M}_k(x) \leq \tau^{\text{share}}\Big\} \tag{2.14}$$

Dataset	$\lvert\mathcal{Y}\rvert$	Train	Dev.	Test	$\mathcal{F}$ test perf.
IMDB	2	20K	5K	25K	94.0
VitaminC	3	370K	10K*	55K	90.6
AG News	4	115K	5K	7.6K	94.4
STS-B	∞	5.7K	1.5K	1.4K	89.8

Table 2.2: Task dataset and label space sizes. The rightmost column reports either test accuracy (classification) or Pearson-correlation (regression). *We downsample the 63K public development set to expedite validation.

Theorem 2.4.6. *For any number of layers $l \in \mathbb{N}^+$, $\mathcal{C}_\epsilon^{\text{share}}(X_{n+1})$ satisfies Eq. (2.6).*

Proof. See Appendix B.1.4 □

2.5 Experimental setup

For our main results, we use an Albert-xlarge model (Lan et al., 2020) with 24 Transformer layers. Results using an Albert-base model and a RoBERTa-large model (Liu et al., 2019) are in Appendix B.3. See Appendix B.2 for implementation details. We did not search across different values for the hyper-parameters of $\mathcal{F}$ or $\mathcal{G}$ as our approach is general and guarantees consistency for any $\mathcal{F}$ with any nonconformity measure. Tuning the hyper-parameters could further improve the efficiency of $\mathcal{G}$.

2.5.1 Tasks

We evaluate on three classification tasks with varying label space size $|\mathcal{Y}|$ and difficulty: **IMDB** (Maas et al., 2011) sentiment analysis on movie reviews, **VitaminC** (Schuster et al., 2021a) fact verification with Wikipedia articles, and **AG** (Gulli, 2004; Zhang et al., 2015) news topic classification. We also evaluate on the **STS-B** (Cer et al., 2017) semantic textual similarity regression task ($\mathcal{Y} = [0, 5]$). Table 2.2 contains dataset statistics, as well as the test result of our original $\mathcal{F}$ model (Albert-xlarge).

2.5.2 Baselines

In addition to our main methods discussed in §2.4.2, we compare to several non-CP baselines. Note that the following methods are not guaranteed to give well-calibrated performance in terms of satisfying Eq. (2.1) (as our CP ones are).

Static. We use the *same* number of layers for all inputs. We choose the exit layer as the first one that obtains the desired consistency on average on $\mathcal{D}_{\text{cal}}$.

Softmax threshold. Following Schwartz et al. (2020b), we exit on the first layer where $p_k^{\max} \geq 1 - \epsilon$, where $p_k^{\max}$ denotes the maximum softmax response of our early classifier. Softmax values are calibrated using temperature scaling (Guo, 2017) on another held-out (labeled) data split, $\mathcal{D}_{\text{scale}}$.

Meta threshold. Even if perfectly calibrated, $p_k^{\max}$ from softmax thresholding is not measuring *consistency* likelihood $\mathbb{P}(\mathcal{G}(X) = \mathcal{F}(X) \mid X = x)$, but rather $\mathbb{P}(\mathcal{G}(X) = Y \mid X = x)$. This is equivalent if $\mathcal{F}$ is an oracle, but breaks down when $\mathcal{F}$ is not. We also experiment with thresholding the confidence value of our meta classifier (§2.3.2) in a similar way (i.e., exiting when it exceeds $1 - \epsilon$).

2.5.3 Evaluation

For each task, we use a proper training, validation, and test set. We use the training set to learn $\mathcal{F}$ and $\mathcal{G}$. We perform model selection on the validation set, and report final numbers on the test set. For all methods, we report the marginalized results over 25 random trials, where in each trial we partition the data into 80% $\mathcal{D}_{\text{cal}}$ $(x_{1:n})$ and 20% $\mathcal{D}_{\text{test}}$ (x_{n+1}). In order to compare different methods across all tolerance levels, we plot each metric as a function of ϵ. Shaded regions show the 16-84th percentiles across trials. We report the following metrics:

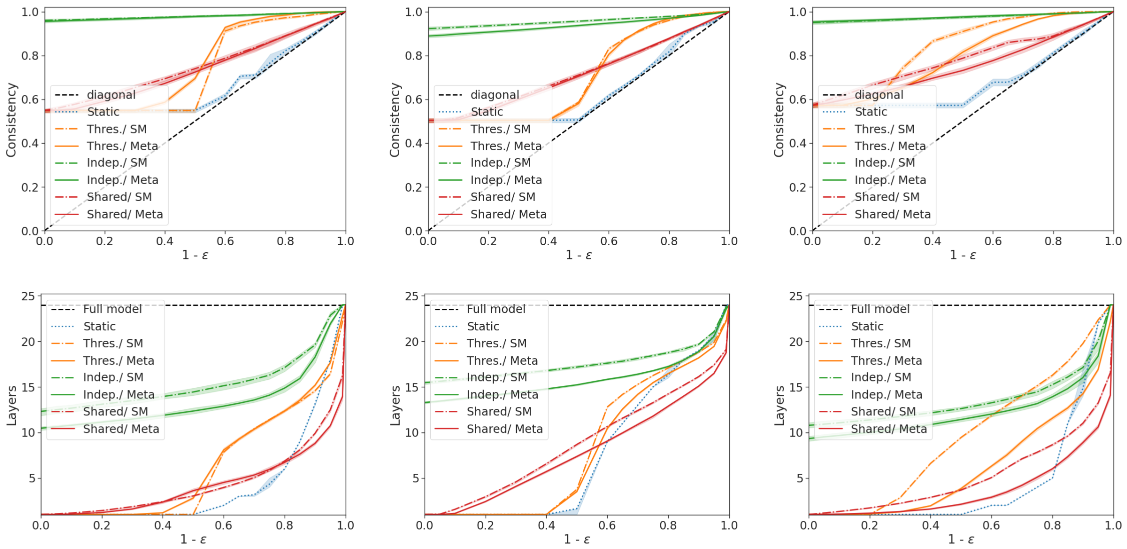
Consistency. We measure the percent of inputs for which the prediction of the CAT model $\mathcal{G}$ is the same as the full Transformer on our test prediction, i.e., $\mathcal{G}(X_{n+1}) = \mathcal{F}(X_{n+1})$. For regression tasks, we count a prediction as consistent if it is within a small margin τ from the reference (we use $\tau = 0.5$). As discussed in §2.1, if $\mathcal{G}$ is ϵ-consistent, we can also derive an average performance lower bound: it will be at least $(1 - \epsilon) \times \mathcal{F}$'s average performance.[8]

Layers (↓). We report the computational cost of the model as the average number of Transformer layers used. Our goal is to improve the efficiency (i.e., use *fewer* layers) while preserving ϵ-consistency. We choose this metric over absolute run-time to allow for implementation-invariant comparisons, but we provide a reference analysis next, to permit easy approximate conversions.

2.5.4 Absolute runtime analysis

The exact run-time of $\mathcal{G}$ depends on the efficiency of the hardware, software, and implementation used. Ideally, the early and meta classifiers can run in parallel with the following Transformer layer (layer $k + 1$). As long as they are faster to compute concurrently than a single layer, this will avoid incurring any additional time cost. For simplicity, however, we use a naïve synchronous implementation. We provide a reference timing for the IMDB task implemented with the Transformers (Wolf et al., 2020) library, PyTorch 1.8.1 (Paszke et al., 2019), and an A100-PCIE-40GB Nvidia GPU with CUDA 11.2 using the naïve approach. A full forward path of an Albert-xlarge takes 22.32ms per input, 0.85ms ×24 for the transformer layers and 1.95ms for the embedding layer and top classifier. Our early classifier takes 0.20ms and the meta classifier takes 0.11ms. Therefore, with a naive implementation, a CAT model $\mathcal{G}$ with an average exit layer less than 17.6 with the meta classifier, or 19.5 without, will realize an overall reduction in wall-clock time relative to the full $\mathcal{F}$. We report example speedup times with the naive implementation in §2.6.3, as well as an implementation agnostic multiply-accumulate operation (MACs) reduction measure. The added computational effort per layer of the early predictor and meta-classifier is marginal (only $66,304$ and $1,920$ MACs, respectively). In comparison, Albert-xlarge with an input length of 256 has $\sim 3 \cdot 10^{11}$ MACs.

[8] In practice, the performance is likely to be *higher* than this lower bound, since inconsistencies with $\mathcal{F}$ could lead to a correct prediction when $\mathcal{F}$ would have otherwise been wrong.



(a) IMDB

(b) VitaminC

(c) AG News

Figure 2-3: Classification results (dev). While both our CP-based methods give valid consistencies (above diagonal), *shared calibration* generally results in earlier exits. This advantage is especially pronounced at smaller tolerance levels (right-hand side), where it significantly outperforms other approaches. Our meta-learned confidence measure $\mathcal{M}_k$ improves over using the softmax response as a drop-in replacement, especially for tasks with larger $|\mathcal{Y}|$. Note that we care more about the right-hand side behavior, (i.e., larger $1 - \epsilon$), as it corresponds to higher consistency.

2.6 Experimental results

We experiment with both our meta classifier $\mathcal{M}_k$ confidence score (**Meta**, §2.3.2), and, for classification tasks, the early classifier's softmax response, $p_k^{\max}$ (**SM**), as a drop-in replacement for $\mathcal{M}_k$ (at no additional cost). Appendix B.3 reports results with other drop-in $\mathcal{M}_k$ replacements, in addition to results using our fixed development set calibration approach (§2.3.3). Appendix B.4 provides qualitative examples.

2.6.1 Classification results

Figure 2-3 summarizes the average consistency and number of layers used by $\mathcal{G}$ as a function of ϵ, while Table 2.3 presents results for specific ϵ on task test sets. Independent calibration proves to be quite conservative due to the loss of statistical power from the loose union bound of the Bonferroni correction for large l (here $l = 24$).

Method	IMDB			VitaminC			AG News		
	Consist.	Acc.	Layers	Consist.	Acc.	Layers	Consist.	Acc.	Layers
$\underline{1 - \epsilon = 0.95}$:		(88.50)			(86.10)			(89.02)	
Static	95.54	92.88	18.36	95.51	89.40	21.00	95.48	93.20	22.00
Thres./ SM	99.65	94.01	16.55	99.83	90.59	20.07	100.00	94.44	22.28
Thres./ Meta	99.98	93.96	17.73	99.73	90.59	19.67	99.41	94.00	16.21
Indep./ Meta	99.66	93.82	15.69	99.07	89.97	19.60	99.81	94.31	20.58
Shared/ SM	97.17	93.24	12.65	96.87	88.99	17.58	97.15	93.43	13.24
Shared/ Meta	97.15	92.71	**10.83**	96.91	89.01	**16.79**	97.08	92.50	**10.17**
$\underline{1 - \epsilon = 0.90}$:		(83.84)			(81.57)			(84.33)	
Static	90.82	89.47	14.00	92.57	87.80	19.00	90.88	89.10	14.00
Thres./ SM	98.88	93.93	14.71	99.05	90.27	18.91	99.68	94.21	19.53
Thres./ Meta	99.75	93.86	15.30	99.10	90.31	18.45	98.90	93.82	13.50
Indep./ Meta	99.39	93.67	14.85	98.29	89.42	18.50	99.60	94.18	17.65
Shared/ SM	94.34	91.77	10.30	93.73	87.00	16.40	94.50	92.01	10.79
Shared/ Meta	94.36	90.78	**9.01**	93.83	86.89	**15.33**	94.29	90.26	**8.35**

Table 2.3: Classification results (test) for specific tolerance levels. We report the accuracy lower bound guaranteed by our CP methods in parentheses. Shared/ Meta is reliably the most efficient method (and is ϵ-consistent). Greyed rows reflect approaches without guarantees; our CAT approaches with guarantees are presented below them.

At some levels of ϵ, non-CP baselines perform competitively, however, they lack formal guarantees. Overall, for the most critical tolerance levels (small ϵ, right-hand side of the plots), our shared method leads to significant efficiency gains while still maintaining the desired level of consistency (above the diagonal).

The effectiveness of our meta predictor, $\mathcal{M}_k$, is most pronounced for tasks with $|\mathcal{Y}| > 2$, where the drop-in softmax score (SM) becomes less indicative of consistency. Both SM and Meta are relatively well-calibrated for IMDB and VitaminC, which makes the threshold-based exit rule a competitive baseline. Still, our Shared/ Meta method provides both reliable and significant gains.

The computational advantage of our CAT model is dependent on the average difficulty of the task and the implementation. As Table 5.2 shows, allowing up to an ϵ of 10% inconsistency, for two of the tasks we cut down the average Transformer layer to only 9 out of 24 using our Shared/ Meta model. This leads to an approximate speedup of 1.8× with a synchronous implementation and of 2.7× with a concurrent one, com-

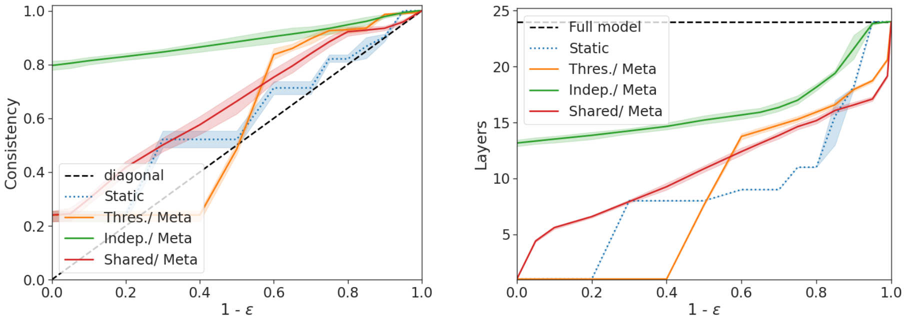


Figure 2-4: Dev results for the STS-B regression task.

	$1-\epsilon = 0.95$		$1-\epsilon = 0.90$	
Method	Consist.	Layers	Consist.	Layers
Static	100.00	24.00	92.51	20.00
Thres./ Meta	99.87	19.19	99.19	18.53
Indep./ Meta	99.29	23.60	97.77	20.26
Shared/ Meta	96.42	**17.64**	92.65	**17.29**

Table 2.4: Test results for the STS-B regression task.

pared to running the full model. Moreover, Figure 2-5 illustrates the user's control over available computational resources via modulating ϵ. Decreasing ϵ increases the confidence level required before committing to the early classifier's prediction (thereby increasing the average number of required layers), and vice-versa.

2.6.2 Regression results

Table 2.4 and Figure 2-4 present results for our regression task, where we see similar trends. Here, an attractive advantage of our meta confidence predictor is its generalizability to multiple task output types. Notice that the event space of $\mathbf{1}\{\mathcal{G}(X) = \mathcal{F}(X)\} = \{0, 1\}$ always, regardless of the original $\mathcal{Y}$.[9] This allows it to be easily adapted to tasks beyond classification, such as regression, where traditional softmax-based confidence measures (as used in, e.g., Schwartz et al. (2020b)) are absent.

[9] As long as equality is suitably defined, e.g., for the STS-B regression task we define consistent outputs as being within distance $\tau = 0.5$ away.

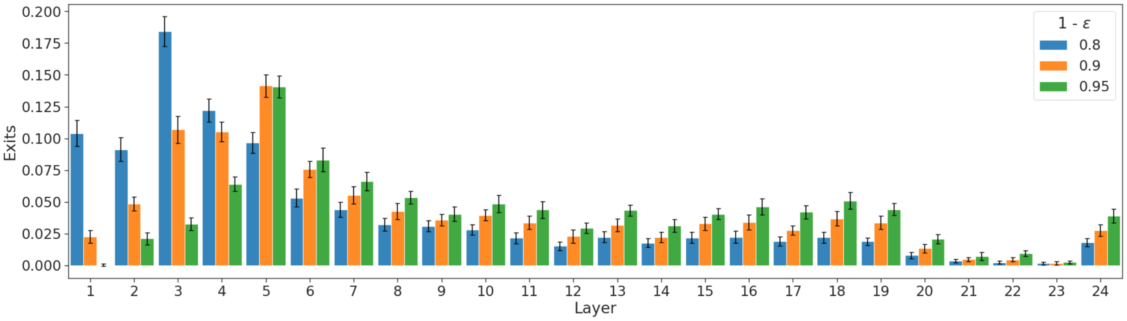


Figure 2-5: Distribution of exit layers per tolerance level ϵ for the IMDB task (dev set) with Shared/ Meta. Larger ϵ allows the CAT model to shift its predictions earlier by permitting for more inconsistencies with the full model $\mathcal{F}$.

Method	**Amortized time** $(100 \cdot T_{\mathcal{G}}/T_{\mathcal{F}})$			
	IMDB	VitaminC	AG News	STS-B
Thres./ SM	76.91	96.66	99.58	N/A
Thres./ Meta	87.22	103.59	77.87	104.01
Indep./ Meta	84.88	103.85	99.44	113.00
Shared/ SM	56.16	**84.86**	58.47	N/A
Shared/ Meta	**54.53** %	87.38 %	**51.10** %	**97.56** %

Table 2.5: Reference time speedup for $1 - \epsilon = 0.90$ (see Table B.3.2 for 0.95). We compute the amortized time with the naive synchronous implementation (§2.5.4).

Method	**MACs reduction** $(\lvert\mathcal{F}\rvert/\lvert\mathcal{G}\rvert)$			
	IMDB	VitaminC	AG News	STS-B
Thres./ SM	1.63	1.27	1.23	N/A
Thres./ Meta	1.57	1.30	1.78	1.30
Indep./ Meta	1.62	1.30	1.36	1.18
Shared/ SM	2.33	1.46	2.22	N/A
Shared/ Meta	×**2.66**	×**1.57**	×**2.87**	×**1.39**

Table 2.6: Reference model complexity reduction for $1 - \epsilon = 0.90$ (see Table B.3.3 for 0.95). The MACs reduction measure is implementation agnostic.

2.6.3 Example efficiency gains

Following the analysis in §2.5.4, we compute the amortized inference time with a naive implementation and report its percentage out of the full model. As Table **??** shows, our Shared calibration is the most efficient method on all four tasks. For tasks with many easy inputs (IMDB and AG News), our Shared/ Meta method can save 45% - 49% of the inference time when $1 - \epsilon = 0.90$. Unsurprisingly, the absolute speedup is less significant for harder tasks, but increases with higher tolerance levels. On VitaminC, even though the Meta measure allows exiting on earlier layers, its additional meta classifiers result in slightly slower inference on average at this tolerance level, compared to our Shared/ SM. With a more efficient concurrent implementation, the Meta measure will be favorable. We also compute the MACs reduction metric which is independent of the specific implementation or hardware and shows the number of multiply-accumulate operations of the full model compared to our CAT model. As demonstrated in Table **??**, our Shared/ Meta method is most effective in reducing the computational effort across all tasks for the two examined tolerance levels.

2.7 Conclusion

The ability to make predictions quickly without excessively degrading performance is critical to production-level machine learning systems. In fact, being capable of quantifying the uncertainty in a prediction and deciding when additional computation is needed (or not) is a key challenge for any intelligent system (e.g., see the System 1 vs. System 2 dichotomy explored in Kahneman (2011)). In this chapter, we addressed the crucial challenge of deciding *when* to sufficiently trust an early prediction of Transformer-based models by learning from their past predictions. Our Confident Adaptive Transformers (CATs) framework leverages *meta predictors* to accurately assess whether or not the prediction of a simple, early classifier trained on an intermediate Transformer representation is likely to already be consistent with that of the *full* model $\mathcal{F}(X)$ (i.e., after all l layers of $\mathcal{F}$ are computed). Importantly, we develop a new conformal prediction approach for calibrating the confidence of the meta classifier that is (1) simple to implement, (2) fast to compute alongside the Transformer, (3) requires only *unlabeled* data, and (4) provides statistically efficient marginal guarantees on the event that the prediction of the faster, amortized CAT model is *consistent* with that of the full $\mathcal{F}$. Our results on multiple tasks demonstrate the generality of our approach, and its effectiveness in consistently improving computational efficiency—all while maintaining a reliable margin of error.

3

Conformal Prediction Cascades

In this chapter, we continue to focus on the efficiency of making reliable predictions. Here we introduce a new approach for conformal prediction (CP) that is well-suited for large-scale, open-ended classification tasks. In the standard CP paradigm, the output sets can often be unusably large and costly to obtain when the correct answer is not unique, and the number of total possible answers is high. We first expand the CP correctness criterion to allow for additional, inferred "admissible" answers, which can substantially reduce the size of the predicted set while still providing valid performance guarantees. Second, we amortize costs by conformalizing prediction cascades, in which we aggressively prune implausible labels early on by using progressively stronger classifiers—again, while still providing valid performance guarantees. We demonstrate the empirical effectiveness of our approach for multiple applications in natural language processing and computational chemistry for drug discovery.

3.1 Introduction

The ability to provide precise performance guarantees is critical to many classification tasks (Amodei et al., 2016; Jiang et al., 2012, 2018). Yet, achieving perfect accuracy with only single guesses is often out of reach due to noise, limited data, insufficient modeling capacity, or other pitfalls. Nevertheless, in many applications, it can be more feasible and ultimately as useful to hedge predictions by having the classifier return a *set* of plausible options—one of which is likely to be correct.

Consider the example of information retrieval (IR) for fact verification. Here the goal is to retrieve a snippet of text of some granularity (e.g., a sentence, paragraph, or article) that can be used to verify a given claim. Large resources, such as Wikipedia, can contain millions of candidate snippets—many of which may independently be able to serve as viable evidence. A good retriever should make precise snippet suggestions, quickly—but do so without excessively sacrificing sensitivity (i.e., recall).

As discussed in earlier sections, conformal prediction (CP) is a methodology for placing exactly that sort of bet on which candidates to retain (Vovk et al., 2005). Concretely, suppose we have been given n examples, $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$, as training data, that have been drawn exchangeably from an underlying distribution P. For instance, in our IR setting, X would be the claim in question, Y a viable piece of evidence that supports or refutes it, and $\mathcal{Y}$ a large corpus (e.g., Wikipedia). Next, let X_{n+1} be a new exchangeable test example (e.g., a new claim to verify) for which we would like to predict the paired $y \in \mathcal{Y}$. The aim of conformal prediction is to construct a set of candidates $\mathcal{C}_\epsilon(X_{n+1})$ likely to contain Y_{n+1} (e.g., the relevant evidence) with *distribution-free marginal coverage* at a tolerance level $\epsilon \in (0, 1)$:

$$\mathbb{P}\left(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})\right) \geq 1 - \epsilon. \tag{3.1}$$

The marginal probability above is taken over all the $n + 1$ calibration and test points $\{(X_i, Y_i)\}_{i=1}^{n+1}$. A classifier is considered to be valid if the frequency of error, $Y_{n+1} \notin \mathcal{C}_\epsilon(X_{n+1})$, does not exceed ϵ. In our IR setting, this would mean including the correct snippet at least ϵ-fraction of the time. Not all valid classifiers, however, are particularly useful (e.g., a trivial classifier that merely returns all possible outputs). A classifier is considered to have good predictive efficiency if $\mathbb{E}[|\mathcal{C}_\epsilon(X_{n+1})|]$ is small (i.e., $\ll |\mathcal{Y}|$). In our IR setting, this would mean not returning too many irrelevant articles—or in IR terms, maximizing precision while holding the level of recall at

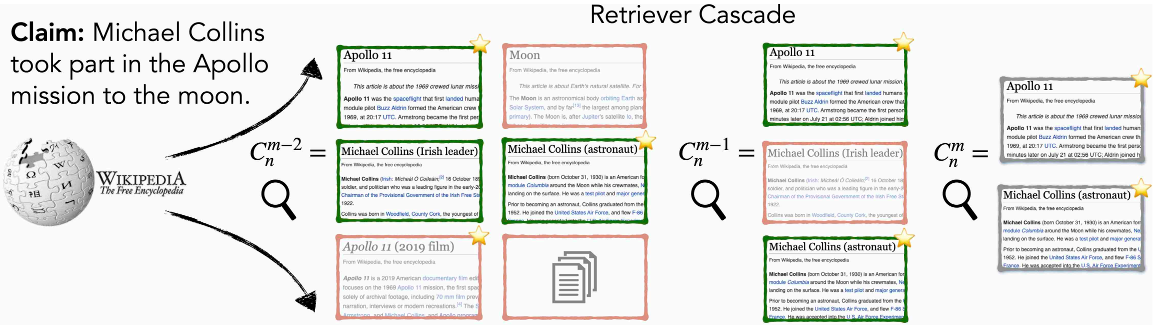

Retriever Cascade
Claim: Michael Collins took part in the Apollo mission to the moon.
WIKIPEDIA
The Free Encyclopedia
$C_n^{m-2} =$
Apollo 11
Moon
Michael Collins (Irish leader)
Michael Collins (astronaut)
Apollo 11 (2019 film)
$C_n^{m-1} =$
Apollo 11
Michael Collins (Irish leader)
Michael Collins (astronaut)
$C_n^{m} =$
Apollo 11
Michael Collins (astronaut)

Figure 3-1: A demonstration of our conformalized cascade with set-valued outputs, here on an IR for claim verification task. The number of considered articles is reduced at every level—red frames are filtered, while green frames pass on. We only care about retrieving *at least one* of the admissible articles (starred) for resolving the claim.

$\geq 1 - \epsilon$ (assuming Y is a single answer). In practice, in domains where the number of outputs to choose from is large and the "correct" one is not necessarily unique, classifiers derived using conformal prediction can suffer dramatically from both poor predictive and computational efficiency (Burnaev and Vovk, 2014; Vovk et al., 2016, 2020). Unfortunately, these two conditions tend to be compounding: large label spaces $\mathcal{Y}$ both (1) often place strict constraints on the set of tractable model classes available for consideration, and (2) frequently contain multiple clusters of labels that are difficult to discriminate between, especially for a low-capacity classifier.

In this chapter, we present two effective methods for improving the efficiency of conformal prediction for classification tasks with large output spaces $\mathcal{Y}$, in which several $y \in \mathcal{Y}$ might be admissible—i.e., acceptable for the purposes of our given task. First, we describe a generalization of Eq. (3.1) to an expanded admission criteria, where $\mathcal{C}_\epsilon(X_{n+1})$ is considered valid if it contains at least one admissible y with high probability. For example, in our IR setting, given the claim *"Michael Collins took part in the Apollo mission to the moon"*, any of the articles *"Apollo 11"*, *"Michael Collins (astronaut)"*, or *"Apollo 11 (2019 film)"* have enough information to independently support it (see Figure 3-1)—and are therefore all admissible. When Y_{n+1} is not unique, forcing the classifier to hedge for the worst case, in which a specific realization of Y_{n+1} must be contained in $\mathcal{C}_\epsilon(X_{n+1})$, is too strict and can lead to conservative predictions. We theoretically and empirically show that optimizing for an expanded admission criteria yields classifiers with significantly better predictive efficiency.

Second, we present a technique for conformalizing *prediction cascades* to progressively filter the number of candidates with a sequence of increasingly complex classifiers. This allows us to balance predictive efficiency with computational efficiency during inference. Importantly, we also theoretically show that, in contrast to other similarly motivated pipelines, our method filters the output space in a manner that still guarantees marginal coverage. Figure 3-1 illustrates our combined approach applied to our IR setting. We demonstrate that, together, these two approaches can serve as complementary pieces of the puzzle towards making conformal prediction more efficient. We empirically validate our approach on information retrieval for fact verification, open-domain question answering, and in-silico screening for drug discovery.

Contributions. In summary, the main results of this chapter are as follows:

- A theoretical extension of validity to allow for inferred *admissible* answers.
- A framework for conformalizing computationally efficient prediction cascades.
- Empirical gains on three diverse tasks with up to 4.6× better predictive efficiency AUC (measured across all ϵ) when calibrating for expanded admission, with pruning factors of up to $1/m$, where m is the number of models, when using cascades.

3.2 Related work

Conformal prediction. As validity is already guaranteed by design in conformal prediction (see §1.3), most efforts in CP focus on improving various aspects of efficiency. Mondrian CP (Vovk et al., 2005) accounts for the fact that some classes are harder to model than others, and leverages class-conditional statistics. Similiarly, several recent studies have built towards conditional—as opposed to marginal—coverage through various adaptive approaches, such as conformalizing quantile functions or working with conditional distributions that vary with x (see Chernozhukov et al., 2019; Kivaranovic et al., 2020; Romano et al., 2019b, 2020a; Cauchois et al., 2021, *inter alia*).

Cauchois et al. (2021) also directly model dependencies among y variables for use in multi-label prediction. Our method for *expanded admission*, on the other hand, aggregates statistics for equivalent *single* labels by example and across classes. Though we only provide marginal guarantees, the ideas expressed in those related works are complementary, and can be applied here as well. Inductive CP (Papadopoulos, 2008), also known as *split* CP, is also complementary extension that dramatically reduces the cost of computing $\mathcal{C}_\epsilon(X_{n+1})$ in the general case; we make use of it here. Most similar to our work, trimmed (Chen et al., 2016) and discretized (Chen et al., 2018) CP trade *predictive* efficiency for *computational* efficiency in regression tasks, where the label space is infinite. A key distinction of our method is that we do not force the same trade-off: in fact, we empirically show that our conformalized cascades can at times result in *better* predictive efficiency alongside a pruned label space.

Prediction cascades. The idea of balancing cost with accuracy by using multi-step inference has been explored extensively for many applications (Jurafsky and Martin, 2000; Fleuret and Geman, 2001; Charniak et al., 2006; Rush and Petrov, 2012; Li et al., 2015; Deng and Rush, 2020). Some of these methods use fixed rules with no performance guarantees, such as greedy pipelines where the top k predictions are passed on to the next level (Chen et al., 2017; Ferrucci et al., 2010). Closer to our work, Weiss and Taskar (2010) optimize their cascades for overall pruning efficiency, and not for top-1 prediction. While they also analyze error bounds for filtering, their guarantees are specific to linear classifiers with bounded L_2 norm, whereas our conformalized approach only assumes data exchangeability. Furthermore, while they assume a target filtering loss before training—our tolerance level ϵ is defined at test time.

3.3 Conformal prediction with expanded admission

We now introduce our strategy for improving the *predictive efficiency* of CP, particularly when the value Y can take to be considered "good enough" is not well-defined. What might it mean for a label y to be "good enough?" Among other factors, this depends on the task, the label space $\mathcal{Y}$, and even the input x. For example, in IR, two different texts might independently provide sufficient information for claim x to be resolved. Formally, we pose the underlying setting as a set-valued function $f : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$, where the ground truth is defined as the expanded set of all *admissible* answers $f(X)$ for input X (e.g., given our notions of semantic equivalence or error tolerance). Unlike ranking or multi-label classification, we only require retrieving a single element of this ground truth set (and without any preference as to which element).

Definition 3.3.1 (Admissible label set)**.** *The underlying (latent) set of labels that are considered to be equally acceptable for input X_i is the admissible label set, $f(X_i)$. Every label in this set is considered substitutable for any other label also in the set.*

It is often the case that this underlying function f remains unknown—after all, exhaustively annotating *all* possible admissible answers can quickly become intractable for many problems. Rather, we assume that what we observe in our dataset are samples, (X_i, Y_i), from the underlying ground truth sets $f(X_i)$ via some additional observation process. For example, often the answer that one sees annotated in a dataset is influenced by which particular annotator wrote it. In this view, the distribution P governing each pair (X_i, Y_i) is an induced distribution from this set-valued function, together with the unknown observation process. We can then use the provided dataset reference Y_i to seed a label-expansion operation, in an attempt to approximate $f(X_i)$. More concretely, for some choice of admission function $g \colon (\mathcal{X} \times \mathcal{Y}) \times \mathcal{Y} \rightarrow \{0, 1\}$, we construct a set of inferred admissible labels $\mathcal{A}_g$ given the seed reference $(X = x, Y = y)$, i.e.,

$$\mathcal{A}_g(x, y) := \Big\{\bar{y} \in \mathcal{Y} \colon g(x, y, \bar{y}) = 1\Big\}. \tag{3.2}$$

We make a few practical assumptions about $\mathcal{A}_g$, namely, that any seed reference that is given to us is also considered admissible, and that $\mathcal{A}_g$ is conservative.

Assumption 3.3.2. *For any $Y \in f(X)$, we have that $\mathcal{A}_g(X, Y)$ contains Y and is a subset of the full ground truth. That is, $\mathcal{A}_g(X, Y)$ obeys $Y \in \mathcal{A}_g(X, Y) \subseteq f(X)$.*

In this work we assume that g is given to us (not learned), and is a deterministic function that has no inherent error in its outputs. For many tasks, this is a quite natural assumption, as it is often feasible for the user to define a set of rules that qualify label admission—e.g., syntactic normalization rules in NLP, or expanding some small δ-neighborhood of the original y given some metric.

Given g, x, and y, any prediction that is a member of the derived set $\mathcal{A}_g$ is then considered to be *admissible*, i.e., a success. For example, in IR for claim verification, let h be the verification model (or a human) that, given the claim x and evidence y, outputs a score for the final verdict (that the claim is true or false). The admission might then be defined as $g(x, y, \bar{y}) := \mathbf{1}\{|h(y, x) - h(\bar{y}, x)| \leq \delta\}$, where δ is a small slack parameter. This then leads us to a helpful definition of expanded admission:

Definition 3.3.3 (Expanded admission)**.** *Given an admission function g and exchangeable calibration examples $\{(X_i, Y_i)\}_{i=1}^n$, a conformal predictor producing admissible predictions $\mathcal{C}_\epsilon(X_{n+1})$ for a new exchangeable test example X_{n+1} is considered to be valid under expanded admission if for any $\epsilon \in (0, 1)$, the set $\mathcal{C}_\epsilon$ satisfies*

$$\mathbb{P}\Big(|\mathcal{A}_g(X_{n+1}, Y_{n+1}) \cap \mathcal{C}_\epsilon(X_{n+1})| \geq 1\Big) \geq 1 - \epsilon. \tag{3.3}$$

Recall that by *predictive efficiency* we mean that $|\mathcal{C}_\epsilon(X_{n+1})|$ should be small. A CP that simply returns all possible labels is trivially valid, but not useful. By Definition 3.3.3, we only need to identify *at least one* $\bar{Y}_{n+1}$ that is deemed admissible according to g. As such, we propose a modification that allows the classifier to be more discriminative—and produce smaller $\mathcal{C}_\epsilon(X_{n+1})$—when testing the null hypothesis that y is not just conforming, but that it is the *most conforming* admissible y for x. Let $\mathcal{N}$ be our non-conformity measure (see §1.3). For each data point $(X_i = x_i, Y_i = y_i)$, we then define the minimal nonconformity score as

$$\mathcal{N}_g^{\min}(x_i, y_i) := \min\Big\{\mathcal{N}(x_i, \bar{y}) \colon \bar{y} \in \mathcal{A}_g(x_i, y_i)\Big\}, \tag{3.4}$$

and use these random variables to compute p-values.

Theorem 3.3.4 (CP with expanded admission)**.** *Assume that* $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$*,* $i = 1, \ldots, n+1$ *are exchangeable. For any non-conformity measure* $\mathcal{N}$*, admission function* g*, and* $\epsilon \in (0,1)$*, define the conformal set at* $x \in \mathcal{X}$ *as*

$$\mathcal{C}_\epsilon^{\min}(x) := \Big\{ y \in \mathcal{Y} \colon \texttt{pval}\left(\mathcal{N}(x,y),\, \{\mathcal{N}_g^{\min}(X_i, Y_i)\}_{i=1}^n\right) > \epsilon \Big\}. \tag{3.5}$$

where `pval` *is defined in Eq.* (1.7)*. Then* $\mathcal{C}_\epsilon^{\min}(X_{n+1})$ *satisfies Eq.* (3.3)*.*

Furthermore, we have that

$$\mathbb{E}\Big[|\mathcal{C}_\epsilon^{\min}(X_{n+1})|\Big] \leq \mathbb{E}\Big[|\mathcal{C}_\epsilon(X_{n+1})|\Big]. \tag{3.6}$$

Proof. See Appendix C.1.1. □

In §3.6 we demonstrate empirically that this simple modification can yield large improvements in efficiency, while still maintaining the desired coverage.

3.4 Conformal prediction with cascaded inference

We now introduce our strategy for improving the *computational efficiency* of conformal prediction. For clarity of presentation, we return to the standard setting without expanded admission, i.e., where $\mathcal{C}_\epsilon(X_{n+1})$ satisfies Eq. (3.1), but emphasize that the method applies to either case. Recall the typical definition of $\mathcal{C}_\epsilon$:

$$\mathcal{C}_\epsilon(x) := \Big\{ y \in \mathcal{Y} \colon \texttt{pval}\Big(\mathcal{N}(x,y), \{\mathcal{N}(X_i, Y_i)\}_{i=1}^n\Big) > \epsilon \Big\}. \tag{3.7}$$

The cost of constructing $\mathcal{C}_\epsilon$ as part of CP is therefore, in general, linear in $|\mathcal{Y}|$. In practice, this can limit the tractable choices available for the nonconformity measure $\mathcal{N}$, particularly in domains where $|\mathcal{Y}|$ is large. Furthermore, predictive and computational efficiency are coupled, as being forced to use weaker $\mathcal{N}$ reduces the statistical power of the CP. This is worse in large-scale applications where computational efficiency is indeed limiting, and where predictive efficiency is particularly important in order for the classifier to be considered useful.

Our approach balances the two by leveraging prediction cascades (Sapp et al., 2010; Weiss and Taskar, 2010), where m models of increasing power are applied sequentially. At each stage, the number of considered outputs is iteratively pruned. Here, we conformalize the cascade, preserving marginal coverage. As we can see above, a nonconformity score and p-value is computed for every candidate $y \in \mathcal{Y}$ when constructing $\mathcal{C}_\epsilon$. Different y, however, might be much easier to reject than others, and can be filtered using simple metrics. For example, in IR, wholly non-topical sentences (of which there are many) can be discarded using fast key-word matching algorithms such as TFIDF or BM25. On the other hand, more ambiguous sentences—perhaps those on the same topic but with insufficient information—might require a more sophisticated scoring mechanism, such as a large neural network.

Assume that we are given a sequence of progressively more discriminative, yet also more computationally expensive, nonconformity measures $(\mathcal{N}_1, \ldots, \mathcal{N}_m)$. When applied in order, we only consider $y \in \mathcal{C}_\epsilon^i(X_{n+1})$ as candidates for $\mathcal{C}_\epsilon^{i+1}(X_{n+1})$, i.e.,

$$\mathcal{C}_\epsilon^m(X_{n+1}) \subseteq \mathcal{C}_\epsilon^{m-1}(X_{n+1}) \subseteq \ldots \subseteq \mathcal{C}_\epsilon^1(X_{n+1}) \subseteq \mathcal{Y}. \tag{3.8}$$

In this way, the amortized cost of evaluating m measures over *parts* of $\mathcal{Y}$ can be lower than the cost of running one expensive measure over *all* of it. For example, in IR, we can use BM25 ($\mathcal{N}_1$) to prune the label space passed to a neural model ($\mathcal{N}_2$). Furthermore, combining multiple nonconformity measures together can also lead to better predictive efficiency when using complementary measures—similar to ensembling (Cherubin, 2019; Linusson et al., 2017; Toccaceli and Gammerman, 2017).

Naïvely applying multiple tests to the same data, however, leads to the *multiple hypothesis testing* (MHT) problem. This results in an increased family-wise error rate (FWER), i.e., false positives, making the CP invalid. Many corrective procedures

Algorithm 2 Cascaded conformal prediction.

Definitions: $(\mathcal{N}_1, \ldots, \mathcal{N}_m)$ is a sequence of nonconformity measures. $\mathcal{M}$ is a monotonic FWER correction. $x_{n+1} \in \mathcal{X}$ is a test point. $x_{1:n} \in \mathcal{X}^n$ and $y_{1:n} \in \mathcal{Y}^n$ are the calibration examples and their labels, respectively. $\mathcal{Y}$ is the label space. ϵ is the tolerance.

```
function PREDICT($x_{n+1}$, $(x_{1:n}, y_{1:n})$, $\epsilon$)
    # Initialize with the full label set.
    $\mathcal{C}_\epsilon^0 \leftarrow \mathcal{Y}$
    # Conservatively set unknown p-values.
    $p_1^{(y)} = p_2^{(y)} = \ldots = p_m^{(y)} \leftarrow 1, \forall y \in \mathcal{Y}$
    for $j = 1$ to $m$ do
        $\mathcal{C}_\epsilon^j \leftarrow \{\}$
        # Iterate through the previous label set.
        for $y \in \mathcal{C}_\epsilon^{j-1}$ do
            # Update the $j$-th p-value for $(x_{n+1}, y)$.
            $p_j^{(y)} \leftarrow \texttt{pval}\left(\mathcal{N}_j(x_{n+1}, y), \{\mathcal{N}_j(x_i, y_i)\}_{i=1}^n\right)$
            # Correct the current p-values for multiple testing.
            $\tilde{p}_j^{(y)} \leftarrow \mathcal{M}(p_1^{(y)}, \ldots, p_m^{(y)})$
            # Keep $y$ only if the corrected p-value is $\leq \epsilon$.
            if $(\tilde{p}_j^{(y)} > \epsilon)$ then
                $\mathcal{C}_\epsilon^j \leftarrow \mathcal{C}_\epsilon^j \cup \{y\}$
    return $\mathcal{C}_\epsilon^m$
```

exist in the literature (e.g., see Liu (1996)). Formally, given m p-values $(P_1, \ldots, P_m)$ for a pair (X, Y), we denote as $\mathcal{M}$ some such correction satisfying

$$\tilde{P} = \mathcal{M}\left(P_1, \ldots, P_m\right) \text{ s.t. } \mathbb{P}\left(\tilde{P} \leq \epsilon \mid Y \text{ is correct}\right) \leq \epsilon. \tag{3.9}$$

Furthermore, we require $\mathcal{M}$ to be element-wise monotonic,[1] i.e.,

$$\left(P_1, \ldots, P_m\right) \preccurlyeq \left(\hat{P}_1, \ldots, \hat{P}_m\right) \Longrightarrow \mathcal{M}\left(P_1, \ldots, P_m\right) \leq \mathcal{M}\left(\hat{P}_1, \ldots, \hat{P}_m\right). \tag{3.10}$$

where $\preccurlyeq$ operates element-wise. We consider several options for $\mathcal{M}$, namely the Bonferroni and Simes corrections (see Appendix C.4). In order to exit the test early at cascade j before all the p-values (i.e., for measures $k > j$) are known, we compute an upper bound for the corrected p-value by conservatively assuming that $P_k = 1$, $\forall k > j$. The procedure is demonstrated in Algorithm 2, and formalized below.

[1] We are unaware of $\mathcal{M}$ beyond contrived examples satisfying Eq. (3.9) but not Eq. (3.10).

Theorem 3.4.1 (Cascaded CP)**.** *Assume that* $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$*,* $i = 1, \ldots, n+1$ *are exchangeable. For any sequence of nonconformity measures* $(\mathcal{N}_1, \ldots, \mathcal{N}_m)$ *yielding p-values* $(P_1, \ldots, P_m)$*, and* $\epsilon \in (0, 1)$*, define the conformal set for step* j *at* $x_{n+1} \in \mathcal{X}$ *as*

$$\mathcal{C}_\epsilon^j(x_{n+1}) := \left\{ y \in \mathcal{Y} \colon \tilde{P}_j^{(y)} > \epsilon \right\}, \tag{3.11}$$

where $\tilde{P}_j^{(y)}$ *is the conservative p-value for* y *at step* j*,* $\mathcal{M}(P_1^{(y)}, \ldots, P_j^{(y)}, 1, \ldots, 1)$*, with* $P_{k>j}^{(y)} := 1$*. Then* $\forall j \in [1, m]$*,* $\mathcal{C}_\epsilon^j(X_{n+1})$ *satisfies Eq. (3.1), and* $\mathcal{C}_\epsilon^m(X_{n+1}) \subseteq \mathcal{C}_\epsilon^j(X_{n+1})$*.*

Proof. See Appendix C.1.2. □

Theorem 3.4.1 also easily extends to the setting of Eq. (3.3). A consequence of this result is that early pruning will *not* affect the validity of the final set, $\mathcal{C}_\epsilon^m$.

3.5 Experimental setup

We empirically evaluate our method on three different tasks with publicly available datasets. In this section, we briefly give a high-level outline of each task and our conformalized approach to it. We also describe our evaluation methodology. We defer the technical details for each task, such as data preprocessing, training, and nonconformity measure formulations, to Appendix C.2.

3.5.1 Tasks

Open-domain question answering (QA). Open-domain question answering focuses on using a large-scale corpus $\mathcal{D}$ to answer arbitrary questions via search combined with reading comprehension. We use the open-domain setting of the Natural Questions dataset (Kwiatkowski et al., 2019). Following Chen et al. (2017), we first retrieve relevant passages from Wikipedia using a document retriever, and then select an answer span from the considered passages using a document reader. We use a Dense Passage Retriever model (Karpukhin et al., 2020) for the retriever, and a BERT model (Devlin et al., 2019) for the reader. The BERT model yields several score variants—we use multiple in our cascade (see Table C.2.1). Any span from any retrieved passage that matches any of the annotated answer strings when lower-case and stripped of articles and punctuation is considered to be correct.

Information retrieval for fact verification (IR). As introduced in §5.1, the goal of IR for fact verification is to retrieve a sentence that can be used to support or refute a given claim. We use the FEVER dataset (Thorne et al., 2018), in which evidence is sourced from a set of ∼40K sentences collected from Wikipedia. A sentence that provides enough evidence for the correct verdict to be determined (whether that is "supports" or "refutes") is considered to be acceptable (multiple are labeled in the dataset). Our cascade consists of (1) a fast, non-neural BM25 similarity score between a given claim and sentence, and (2) the score of an ALBERT model (Lan et al., 2020) trained to directly predict if a given claim and sentence are related.

In-silico screening for drug discovery (DR). In-silico screening of chemical compounds is a common task in drug discovery and drug repurposing, where the goal is to identify possibly effective drugs to manufacture and test (Stokes et al., 2020). Using the ChEMBL database (Mayr et al., 2018), we consider the task of screening molecules for combinatorial constraint satisfaction, where given a specified constraint such as "*has property A but not property B*", we want to identify at least one molecule from a given set of candidates that has the desired attributes. Our cascade consists of (1) the score of a fast, non-neural Random Forest (RF) applied to binary Morgan fingerprints (Rogers and Hahn, 2010), and (2) the score of a directed Message Passing NN (MPNN) ensemble trained using `chemprop` (Yang et al., 2019).

3.5.2 Evaluation metrics

For each task, we use a proper training, validation, and test set. We use the training set to learn all nonconformity measures. We perform model selection specifically for CP on the validation set, and report final numbers on the test set. For all CP methods, we report the marginalized results over 20 random trials, where in each trial we partition the data into 80% calibration ($x_{1:n}$) and 20% prediction points (x_{n+1}).

In order to compare the aggregate performance of different CPs across all tolerance levels, we plot each metric as a function of ϵ, and compute the area under the curve (AUC). In all plots, shaded regions show the 16-84th percentiles across trials.

For evaluation, we use the following metrics:

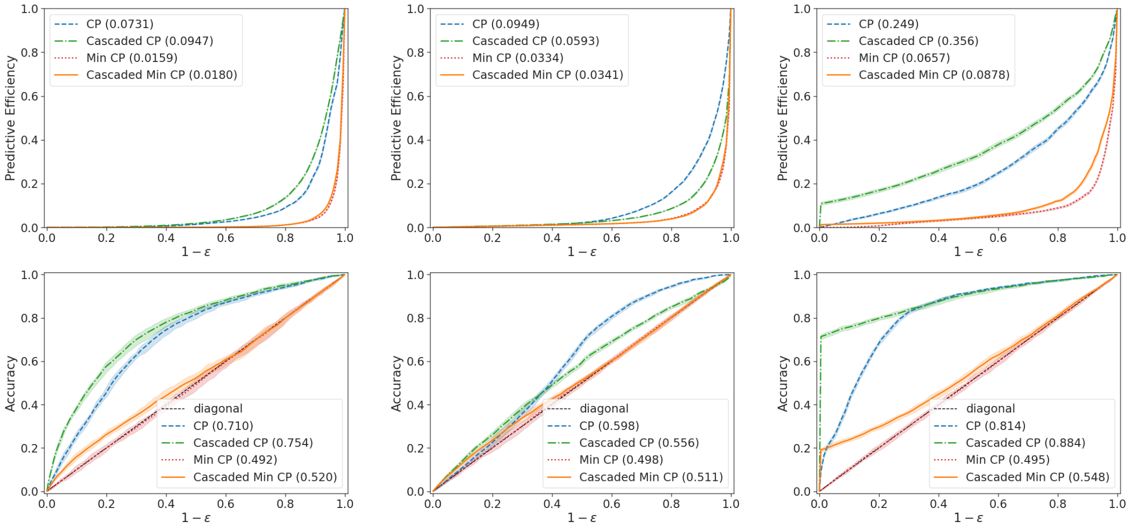
Predictive accuracy. We measure accuracy as the rate at which at least one admissible prediction is in $\mathcal{C}_\epsilon$, i.e., $|\mathcal{A}_g(X_{n+1}, Y_{n+1}) \cap \mathcal{C}_\epsilon(X_{n+1})| \geq 1$. To be valid, the key criteria in this work, classifier should have an accuracy rate $\geq 1 - \epsilon$, and AUC ≥ 0.5. Note that *more* is <u>not</u> necessarily *better*: higher success rates than required can lead to poor efficiency (i.e., the size of $\mathcal{C}_\epsilon$ can afford to decrease at the expense of accuracy).

Predictive efficiency (↓). We measure predictive efficiency as the size of the prediction set out of all candidates: $|\mathcal{C}_\epsilon| \cdot |\mathcal{Y}|^{-1}$. The goal is to make the predictions more precise while still maintaining validity. *Lower* predictive efficiency is better (↓), as it means that the size of $\mathcal{C}_\epsilon$ is relatively *smaller*.

Amortized computation cost (↓). We measure the amortized computation cost as the ratio of `pvalue` computations required to make a cascaded prediction with early pruning, compared to when using a simple combination of CPs (no pruning, same number of measures). In this work we do not measure wall-clock times as these are hardware-specific, and depend heavily on optimized implementations. *Lower* amortized cost is better, as it means that the relative number of p-value computations required to construct $\mathcal{C}_\epsilon^m$ (for a given m) is *smaller*.

3.6 Experimental results

In the following, we address several key research questions relating to our two combined conformal prediction advancements, and their impact on overall performance. In all of the following experiments, we report results on the test set, using cascade configurations selected based on the validation set performance. The QA and IR cascades use the Simes correction for MHT, while the DR cascades uses the Bonferroni correc-



(a) QA (b) IR (c) DR

Figure 3-2: Predictive efficiency and success rates as a function of ϵ. The efficiency of the *min*-calibrated CPs (both cascaded and non-cascaded) is significantly better (i.e., lower) than standard CP across all tasks, especially at critical values of ϵ. The *min*-calibrated CP's accuracy hugs the diagonal, allowing it to remain valid, but be far less conservative (resulting in better efficiency).

tion. Additional results, details, and analysis are included in Appendix C.3.

Predictive efficiency of expanded admission. We begin by testing how deliberately calibrating for expanded admission affects predictive efficiency. That is, we use minimal nonconformity scores $\{\mathcal{N}_g^{\min}(X_i, Y_i)\}_{i=1}^n$ for calibration (Eq. 3.4) to create $\mathcal{C}_\epsilon(X_{n+1})$ using Eq. 3.5. Figure 3-2 shows the predictive efficiency of our *min*-calibrated CPs with expanded admission across all ϵ, while Table 3.1 shows results for select values. We compare both cascaded and non-cascaded CPs, as Theorem 3.3.4 applies to both settings. The non-cascaded CP uses the last nonconformity measure of the cascaded CP. Across all tasks, the efficiency of the *min*-calibrated CP is significantly better than the baseline CP method, and results in tighter prediction sets—giving up to 4.6× smaller AUC. Naturally, this effect depends on the qualities of the admission function g and resulting admissible label sets $\mathcal{A}_g$. For example, the most dramatic gains are seen on QA. This task has a relatively large variance among admissible label nonconformity scores (i.e., some admissible answer spans are much

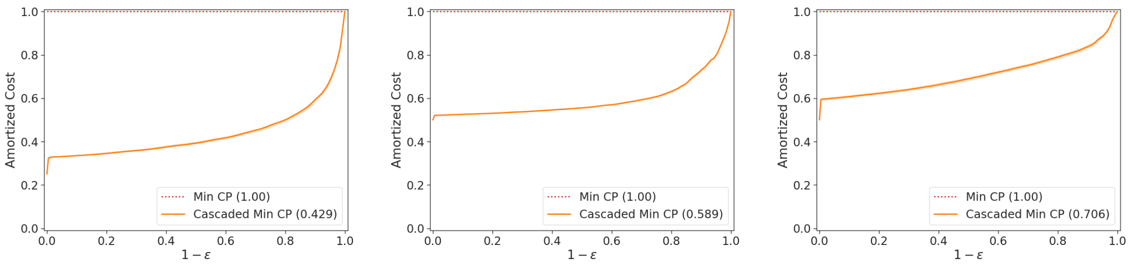
Task	Target Acc.	Baseline CP		*min*-CP		Cascaded *min*-CP		
	$(1-\epsilon)$	Acc.	$\lvert\mathcal{C}_\epsilon\rvert$	Acc.	$\lvert\mathcal{C}_\epsilon^{\min}\rvert$	Acc.	$\lvert\mathcal{C}_\epsilon^{\min}\rvert$	Amortized cost
QA	0.90	0.98	1245.7	0.90	198.0	0.90	235.1	0.59
	0.80	0.94	453.5	0.80	58.7	0.80	57.7	0.50
	0.70	0.91	227.8	0.70	22.1	0.70	20.6	0.45
	0.60	0.87	127.8	0.60	10.6	0.61	9.5	0.42
IR	0.99	1.00	33.6	0.99	17.8	0.99	18.0	0.92
	0.95	1.00	20.9	0.95	7.4	0.95	7.4	0.79
	0.90	0.98	13.8	0.90	4.0	0.90	3.8	0.73
	0.80	0.95	6.7	0.80	1.7	0.80	1.6	0.63
DR	0.90	0.99	429.8	0.90	84.1	0.91	147.8	0.84
	0.80	0.97	305.8	0.80	42.7	0.81	62.1	0.79
	0.70	0.96	216.6	0.70	28.1	0.71	37.3	0.75
	0.60	0.94	145.6	0.60	19.3	0.63	24.6	0.72

Table 3.1: CP results for specific tolerance levels ϵ. Each line shows the empirical accuracy (Acc.) and the (raw) size of the prediction set for our two methods as compared to regular CP, per target accuracy rate $(1-\epsilon)$. The amortized computation cost of the cascade, a consequence of pruning, is also given.

easier to identify than others), and thus calibrating for the most "conforming" spans has a large impact.

Validity of minimal nonconformity calibration. As per Theorem 3.3.4, we observe that our *min*-calibrated CP still creates valid predictors, with an accuracy rate close to $1-\epsilon$ on average. We see that the *min*-calibration allows the predictor to reject more wrong predictions (lower predictive efficiency, which is better) while, with high enough probability, still accept at least one that is correct. The standard CP methods, however, are more conservative—and result in prediction sets and accuracy rates larger than necessary. This effect is most pronounced at smaller ϵ (e.g., < 0.2).

Computational efficiency of conformalized cascades. Figure 3-3 shows the amortized cost, in terms of percentage of p-value computations skipped, achieved using our cascaded CP algorithm. Our method reduces the number of p-value computations by up to a factor of $1/m$ for an m-layer cascade. The effect of conformalization is clear: the more strict an ϵ we demand, the fewer labels we can prune, and vice versa. To simplify comparisons, our metric gives equal weight to all p-value computations. In practice, however, the benefits of early pruning will generally grow by layer, as the later p-values



(a) QA

(b) IR

(c) DR

Figure 3-3: Pruning rates as function of ϵ. Incorporating early rejection via the cascade reduces the fraction of required p-value computations. Larger tolerance levels (ϵ) allow for more aggressive pruning.

are assumed to be increasingly expensive to compute. Again, exactly quantifying this trade-off in terms of absolute cost is model- and implementation-specific.

Conservativeness of conformalized cascades. A limitation of the cascade approach is the conservative nature of the corrected p-values, which can reduce the statistical power of the CP as the number of cascade layers grows. This effect is especially present if the cascaded measures are highly dependent. In general, however, in both Figure 3-2 and Table 3.1, we see that the benefits of combining complementary cascaded models largely make up for this drop in statistical power, as our cascaded *min*-calibrated CPs nearly matches the predictive efficiency of our non-cascaded models. Importantly, this is achieved while still improving computational efficiency.

Relation to heuristic methods. For completeness, in Appendix C.3.2 we also compare CP to common heuristic methods for producing set-valued predictions—namely, taking the top-k predictions and taking all predictions for which the model's score exceeds a threshold τ. We show that while CP is more general, it can be (practically) reduced to each of these methods with the appropriate choice of nonconformity measure. In some cases, the flexibility of CP even allows for better predictive efficiency, even while those heuristics do not amortize cost or guarantee coverage in finite samples.

3.7 Conclusion

Conformal prediction can afford remarkable theoretical performance guarantees to applications for which high accuracy and precise confidence estimates are key. Naïvely applying CP, however, can be inefficient in practice. This is especially true in domains in which the correct answers are not clearly delineated, and in which the computational cost of discriminating between options starts to become a limiting factor. In this chapter, we proposed two novel methods that provide two more pieces of the puzzle. Our results show that (1) calibration using expanded admission consistently improves empirical predictive efficiency, and (2) conformal prediction cascades yield better computational efficiency—thereby enabling the use of more powerful classifiers.

4

Constrained Conformal Prediction

In this chapter, we develop a new approach to conformal prediction with constraints. Standard conformal prediction provides the ability to adapt to model uncertainty by constructing a calibrated candidate set with guarantees that the set contains the correct answer with high probability. In order to obey this coverage property, however, conformal sets can become inundated with noisy candidates—which can render them unhelpful in practice. This is particularly relevant to practical applications where there is a limited budget, and the cost (monetary or otherwise) associated with false positives is non-negligible. We propose to trade coverage for a notion of precision by enforcing that the total number of false positives in the predicted conformal sets is bounded according to a user-specified tolerance. Subject to this constraint, our algorithm then optimizes for a generalized notion of set coverage (i.e., the true positive rate) that allows for any number of true answers for a given query (including zero). We demonstrate the effectiveness of this approach across a number of classification tasks in natural language processing, computer vision, and computational chemistry.

4.1 Introduction

Conformal prediction (Vovk et al., 2005) is remarkable in the sense that it yields confident prediction sets that provably contain the correct answer with high proba-

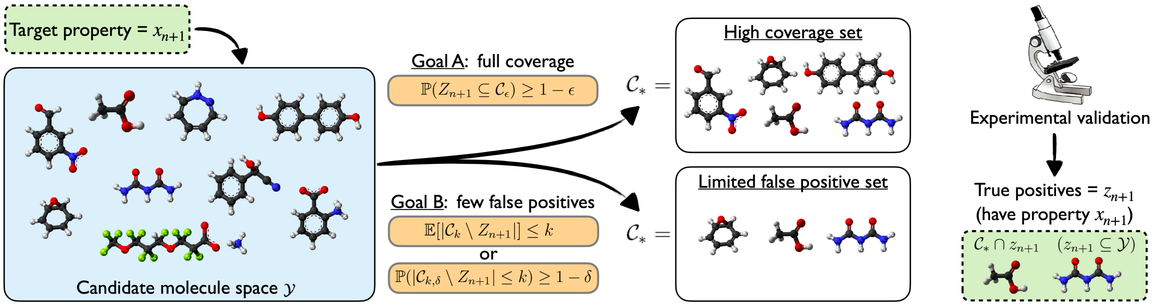

Target property = x_{n+1}
Candidate molecule space $\mathcal{Y}$
Goal A: full coverage
$\mathbb{P}(Z_{n+1} \subseteq \mathcal{C}_\epsilon) \geq 1 - \epsilon$
Goal B: few false positives
$\mathbb{E}[|\mathcal{C}_k \setminus Z_{n+1}|] \leq k$
or
$\mathbb{P}(|\mathcal{C}_{k,\delta} \setminus Z_{n+1}| \leq k) \geq 1 - \delta$
$\mathcal{C}_* =$
High coverage set
$\mathcal{C}_* =$
Limited false positive set
Experimental validation
True positives = z_{n+1}
(have property x_{n+1})
$\mathcal{C}_* \cap z_{n+1}$
$(z_{n+1} \subseteq \mathcal{Y})$

Figure 4-1: A demonstration of our approach to relaxing standard coverage guarantees ("Goal A") in favor of rigorous limits on the total number of false positives included in the output $\mathcal{C}_{k,\delta}$ ("Goal B"). In the case of in-silico screening for drug discovery, limiting false positives is critical when balancing a budget for experimental validation.

bility. For many classification problems, these guarantees make the prediction sets useful tools for analyzing uncertainty (Gammerman and Vovk, 2007; Lei, 2014; Romano et al., 2020c). Unfortunately, however, these guarantees do not come for free. Rather, in order to achieve proper coverage on difficult tasks, conformal prediction can often be unable to rule out an overwhelming number of candidates, making their prediction sets large and inefficient. This can make conformal predictors unusable in settings in which the cost of returning false positive predictions is substantial.

As an example, consider in-silico screening for drug discovery (see Figure 4-1). In-silico screening uses computational tools to search over millions of molecular compounds to identify candidates with desired properties. Any identified candidates are then verified experimentally. While it is often not necessary to return *all* possible viable candidates (e.g., even identifying just one effective drug can suffice), it is important to respect budgetary constraints by avoiding false positive predictions. Too many false positives can quickly consume available resources (e.g., time, materials, funding, or other assets). This is especially relevant when a valid answer—which in this case is an effective drug—might not even exist (a common occurrence)

In this chapter, we develop an approach to creating confident prediction sets that trades off standard coverage guarantees for constraints on the total number of false positives (FP). In other words, we shift the focus of our conformal guarantees to be on limiting the number of incorrect answers in our outputs, with the understanding that we can potentially fail to recover some proportion of the true answers, i.e., we may obtain a lower true positive rate (TPR), which we assume is acceptable.

Concretely, we are interested in a set prediction setting where we have been given n multi-label classification examples $(X_i, Z_i) \in \mathcal{X} \times 2^{\mathcal{Y}}$, $i = 1, \ldots n$ as calibration data, that have been drawn exchangeably from some underlying distribution P_{XZ}. Under our assumptions, each observation X_i can be associated with any number of correct labels (including zero, in the case of having no answer at all, or one, like standard classification). That is, the response variable Z_i is a subset of the full label space $\mathcal{Y}$. For example, in the above in-silico screening task, X_i would be the current property being screened for, $\mathcal{Y}$ the space of all molecular candidates that might have this property, and $Z_i \subseteq \mathcal{Y}$ the set of molecules that do have it. Let $X_{n+1} \in \mathcal{X}$ be a new exchangeable test example for which we would like to predict the set of correct labels, $Z_{n+1} \subseteq \mathcal{Y}$. Our goal is to construct a set predictor $\mathcal{C}_k(X_{n+1})$ that maximizes recall of Z_{n+1} (i.e., TPR), while limiting the expected number of false positives according to a user-defined tolerance $k \in \mathbb{R}_{>0}$:

$$\text{maximize } \mathbb{E}\left[\frac{|\mathcal{C}_k(X_{n+1}) \cap Z_{n+1}|}{\max(|Z_{n+1}|, 1)}\right] \quad \text{s.t.} \quad \mathbb{E}\Big[|\mathcal{C}_k(X_{n+1}) \setminus Z_{n+1}|\Big] \leq k. \tag{4.1}$$

As an alternative to bounding the expected number of false positives, we can also seek a predictor $\mathcal{C}_{k,\delta}$ that controls the probability of exceeding k false positives:

$$\text{maximize } \mathbb{E}\left[\frac{|\mathcal{C}_{k,\delta}(X_{n+1}) \cap Z_{n+1}|}{\max(|Z_{n+1}|, 1)}\right] \quad \text{s.t.} \quad \mathbb{P}\Big(|\mathcal{C}_{k,\delta}(X_{n+1}) \setminus Z_{n+1}| \leq k\Big) \geq 1 - \delta, \tag{4.2}$$

where $\delta \in (0, 1)$ is another user-defined tolerance level. Both constructions define different, but useful, operating conditions; the first is more straightforward (e.g., for the general practitioner), while the second offers a finer level of control. Note that both constraints are marginal over the choice of calibration and test data.

In order to achieve the desired levels of false positive control, we present an approach that is based on *set classification*, combined with conformal calibration techniques (Shafer and Vovk, 2008; Papadopoulos, 2008; Alvarsson et al., 2021). Specifically, we use a set nonconformity measure $\mathcal{F}\colon \mathcal{X} \times 2^{\mathcal{Y}} \to \mathbb{R}$ to score candidate output sets, $\mathcal{S} \in 2^{\mathcal{Y}}$, for a given input $x \in \mathcal{X}$. Intuitively, a *high* nonconformity score (e.g., loss) should reflect the confidence that the candidate set might contain a *high* number of false positives, and vice-versa. We learn this function from separate multi-label classification training data. As enumerating and scoring all possible candidate sets is combinatorially hard, we instead adopt the nested conformal prediction strategy of Gupta et al. (2019), where we greedily construct prediction sets using a best-first strategy that adds top-ranked individual labels to a growing, nested output set $\mathcal{S}$. We stop when its nonconformity score, $\mathcal{F}(x, \mathcal{S})$, exceeds a calibrated threshold—that we find based on our desired false positive constraints. This greedy approach both allows us to scale to larger label spaces $\mathcal{Y}$ (i.e., where there are many candidate labels that choose from when composing the prediction set), and to leverage existing statistical theory for calibrating expectations of monotonic losses for nested set predictors (Gupta et al., 2019; Bates et al., 2020; Angelopoulos et al., 2022a).

Contributions. In summary, the main results of this chapter are as follows:

- An adaptation of conformal prediction that provides rigorous false positive control;
- A simple strategy for constructing output sets with high true positive rates;
- An empirical demonstration of the utility of our method across classification tasks in NLP, computer vision, and computational chemistry.

4.2 Related work

Conformal prediction. As introduced in §1.3, conformal prediction provides a finite-sample, distribution-free method for obtaining prediction sets $\mathcal{C}$ with guarantees on the event $\mathbf{1}\{Y_{n+1} \in \mathcal{C}(X_{n+1})\}$. Most efforts in CP focus on improving the predictive efficiency, $\mathbb{E}[|\mathcal{C}(X_{n+1})|]$, of the conformal sets (Vovk et al., 2016; Sadinle et al., 2019; Romano et al., 2020c; Angelopoulos et al., 2021b; Fisch et al., 2021a,c; Hoff, 2021). As coverage is guaranteed by design, improving efficiency will naturally lead to more precise sets with fewer false positives—but not to a specifiable level. Cauchois et al. (2021) develop a conformal approach to multi-label classification that can guarantee that the prediction set only contains true labels (i.e., FP $= 0$), but does not offer fine-grained control. Most relevant to our work, Bates et al. (2020) develop a flexible framework for controlling the risk, $\mathbb{E}[\mathcal{L}(Y, \mathcal{T}(X))]$, of a set-valued predictor $\mathcal{T}$ with an arbitrary loss function $\mathcal{L}$—as long the loss respects a monotonic *nesting* property, $\mathcal{S} \subset \mathcal{S}' \Rightarrow \mathcal{L}(\mathcal{S}) \geq \mathcal{L}(\mathcal{S}')$, for any two prediction sets $\mathcal{S}$ and $\mathcal{S}'$. The calibration strategy we use here for marginal expectations is based on an extension in Angelopoulos et al. (2022a). Angelopoulos et al. (2021a) proposed methods to rigorously control non-monotonic losses, including the related false discovery rate (FDR), which normalizes the number of false positives over the size of the prediction set. However, as most of our target applications have relatively few true positives, FDR control can lead to many empty predictions, which makes controlling total false positives a more natural fit for this work (see also our extended discussion in Appendix D.5). Finally, though we focus on conformal approaches, our methods are tightly connected to the broader literature surrounding distribution-free calibration (Vovk et al., 2004, 2015; Vovk and Petej, 2014; Gupta et al., 2019, 2020a; Barber, 2020).

Multiple hypothesis testing. Controlling the number of false positives/discoveries over a collection of hypothesis tests is well-studied (Dunn, 1961; Benjamini and Hochberg, 1995; Lehmann and Romano, 2005a; Romano and Wolf, 2007). In fact, the objectives expressed in Eqs. (4.1) and (4.2) are established concepts in statistics—i.e., PFER, the per-family error rate, and k-FWER, the generalized familywise error rate (Spjøtvoll, 1972; Romano and Wolf, 2007). Recently, FDR control has also been studied for outlier detection in a conformal inference setting (Bates et al., 2021). Classic approaches operate over p-values for each hypothesis test that have specific dependency structures (e.g., independent or positively dependent), or otherwise use more conservative corrections. Though similar, our multi-label setting is slightly different from standard multiple testing in that there is both (1) an unknown dependency structure between candidate labels for the same query, but also (2) an extra layer of exchangeability over the $n+1$ queries. Our approach is able to ignore (1) by leveraging (2) within a conformal calibration framework.

Selective classification. In selective classification (El-Yaniv and Wiener, 2010), models can abstain from answering. In particular, Geifman and El-Yaniv (2017) propose a strategy for finding classifiers with specific selective 0/1 risks (i.e., the expected accuracy over *answered* examples). In our setting, this is analogous to controlling false positives using $k \approx 0$. If uncertain, the model would have to "abstain" by outputting an empty set. Our framework also generalizes this behavior for any positive k.

4.3 Set predictions with limited false positives

We now introduce our strategy for limiting the number of false positives that are contained in our output sets. Once again, we assume that we have been given n exchangeable multi-label classification examples, $(X_i, Z_i) \in \mathcal{X} \times 2^{\mathcal{Y}}$, $i = 1, \ldots n$ as calibration data, that are drawn from a distribution P_{XZ}. We follow split conformal prediction, and assume that any training data used is distinct from this calibration data. The response Z_i is treated as a generalized set of correct labels for input X_i, and is a subset of $\mathcal{Y}$. For example, in the in-silico screening task from §5.1, X_i is the current target property being screened for, $\mathcal{Y}$ is the space of all molecular candidates, and $Z_i \subseteq \mathcal{Y}$ is the set of molecules that have that property.

For a set $\mathcal{C}(x) \subseteq \mathcal{Y}$ evaluated at a point $x \in \mathcal{X}$ with labels $z \subseteq \mathcal{Y}$, we define the *true positive proportion* (TPP) as the ratio of correct labels that are recovered:

$$\mathrm{TPP}(z, \mathcal{C}(x)) := \frac{|\mathcal{C}(x) \cap z|}{\max(|z|, 1)} \tag{4.3}$$

(note that TPR := $\mathbb{E}[\mathrm{TPP}]$), and the number of *false positives* (FP) as the total count of incorrect labels in $\mathcal{C}(x)$:

$$\mathrm{FP}(z, \mathcal{C}(x)) := |\mathcal{C}(x) \setminus z|. \tag{4.4}$$

Our goal, as stated in §5.1, is to maximize the expected TPP, while constraining the FP in either of two ways:

Definition 4.3.1 (k-FP validity)**.** *A conformal classifier producing test prediction $\mathcal{C}_k(X_{n+1})$ is k-FP valid if it satisfies $\mathbb{E}[\mathrm{FP}(Z_{n+1}, \mathcal{C}_k(X_{n+1}))] \leq k$.*

Definition 4.3.2 ((k, δ)-FP validity)**.** *A conformal classifier producing test prediction $\mathcal{C}_{k,\delta}(X_{n+1})$ is (k, δ)-FP valid if it satisfies $\mathbb{P}(\mathrm{FP}(Z_{n+1}, \mathcal{C}_{k,\delta}(X_{n+1})) \leq k) \geq 1 - \delta$.*

4.3.1 An oracle set predictor

To motivate our approach, imagine an *oracle* with access to $P_{Z|X}$, the conditional distribution of the multi-label set Z given the input X. Given this information, for any input $x \in \mathcal{X}$ and candidate set $\mathcal{S} \in 2^{\mathcal{Y}}$, in theory such an oracle would be able to exactly calculate both the expectation and the conditional distribution of the number of false (and true) positives in $\mathcal{S}$ given x. In order to maximize the TPR while meeting k-FP and (k, δ)-FP validity, it could then yield:

$$\mathcal{C}_k^{\mathrm{oracle}}(x) := \underset{\mathcal{S} \in 2^{\mathcal{Y}}}{\arg\max} \Big\{ \mathbb{E}[\mathrm{TPP}(Z, \mathcal{S}) \mid x] \colon \mathbb{E}[\mathrm{FP}(Z, \mathcal{S}) \mid x] \leq k \Big\} \tag{4.5}$$

$$\mathcal{C}_{k,\delta}^{\mathrm{oracle}}(x) := \underset{\mathcal{S} \in 2^{\mathcal{Y}}}{\arg\max} \Big\{ \mathbb{E}[\mathrm{TPP}(Z, \mathcal{S}) \mid x] \colon \mathbb{P}(\mathrm{FP}(Z, \mathcal{S}) \mid x] > k) < \delta \Big\} \tag{4.6}$$

where ties are settled by smaller set size. Of course, computing this oracle is not possible, as $P_{Z|X}$ is unknown. Furthermore, enumerating all sets $\mathcal{S} \in 2^{\mathcal{Y}}$ is infeasible for large $\mathcal{Y}$. Instead, in the following sections we develop a practical approach for roughly approximating the oracle's behavior with three main components:

1. A **set function** $\mathcal{F} \colon \mathcal{X} \times 2^{\mathcal{Y}} \to \mathbb{R}$ that directly generates a score for a candidate set $\mathcal{S}$ given x that is predictive of either $\mathbb{E}[\mathrm{FP}(Z, \mathcal{S}) \mid x]$ or $\mathbb{P}(\mathrm{FP}(Z, \mathcal{S}) \mid x] > k)$.

2. A **calibrated search strategy** for exploring a tractable number of candidate sets, and identifying valid sets satisfying our constraints using predictions from $\mathcal{F}$;

3. A **selection policy** for picking a final output set.

Algorithm 3 Training the LikelihoodModel p_θ and SetModel $\mathcal{F}$ (§4.3.2).

Definitions: $\mathcal{D}_{\text{cal}}$ is a calibration set. LikelihoodModel is an abstract model that estimates individual label likelihood for ranking and set item featurization. SetModel is an abstract model that estimates FP (we use DeepSets).

```
function TRAIN($\mathcal{D}_{\text{train}}$)
    # Split the training data into two disjoint sets.
    $\mathcal{D}^{(1)}_{\text{train}}, \mathcal{D}^{(2)}_{\text{train}} \leftarrow$ SPLIT($\mathcal{D}_{\text{train}}$)
    # Use one set to estimate individual likelihoods, $p_\theta(y_c \in Z \mid x)$.
    $p_\theta(y_c \in Z \mid x) \leftarrow$ FIT(LikelihoodModel, $\mathcal{D}^{(1)}_{\text{train}}$)
    # Use the other (smaller) set to learn the FP set function, $\mathcal{F}(x, \mathcal{S})$.
    $\mathcal{F}(x, \mathcal{S}) \leftarrow$ FIT(SetModel, $p_\theta$, $\mathcal{D}^{(2)}_{\text{train}}$)
    return $p_\theta$, $\mathcal{F}$
```

Wherever possible, our proposed method will try to balance simplicity and efficiency with effectiveness. Theoretically, however, the framework it follows is model-agnostic.

4.3.2 Scoring candidate sets with set functions

We choose to model $\mathcal{F}$ using DeepSets (Zaheer et al., 2017). DeepSets is a popular method which is known to be a universal approximator for continuous set functions, which makes it a natural choice for our purpose. Let $\{\phi(x, y_1), \ldots, \phi(x, y_{|\mathcal{S}|})\}$ featurize a candidate set $\mathcal{S} \subseteq \mathcal{Y}$, where $\phi(x, y_c) \in \mathbb{R}^d$ is a function of (x, y_c), for $y_c \in \mathcal{S}$. In practice, we find that taking $\phi(x, y_c)$ to be an estimate of $p_\theta(y_c \in Z \mid x)$, the marginal likelihood of y_c being a correct label, performs well and is simple to implement. These one-dimensional prediction scores can be provided by any base model.[1] For example, in our in-silico screening task, we define ϕ using a directed MPNN (Yang et al., 2019) that independently predicts the probability of an individual molecule having the properties targeted by the screen, or not. Given ϕ, the DeepSets model is defined by

$$\Psi(x, \mathcal{S}) := \text{softmax}\Big(\text{dec}\Big(\sum_{y_c \in \mathcal{S}} \text{enc}(\phi(x, y_c))\Big)\Big), \tag{4.7}$$

where enc$(\cdot)$ and dec$(\cdot)$ are neural encoder and decoder models, and softmax$(\cdot)$ is taken over the range of possible false positives, $\{0, \ldots, |\mathcal{S}|\}$. Ψ is trained to predict

[1] This is comparable to the 1-d features used by Platt scaling.

the total number of false positives in $\mathcal{S}$ via cross entropy, using labeled sets sampled from held-out training data, separate from the split used to learn p_θ (used for ϕ). We then compute $\mathcal{F}_k$ and $\mathcal{F}_{k,\delta}$ (for either k-FP or (k, δ)-FP validity) as

$$\mathcal{F}_k(x, \mathcal{S}) := \sum_{\eta=0}^{|\mathcal{S}|} \eta \cdot \Psi(x, \mathcal{S})_\eta \tag{4.8}$$

$$\mathcal{F}_{k,\delta}(x, \mathcal{S}) := 1 - \sum_{\eta=0}^{\min(k,|\mathcal{S}|)} \Psi(x, \mathcal{S})_\eta, \tag{4.9}$$

where $\Psi(x, \mathcal{S})_\eta$ denotes the η-th index of the softmax (i.e., the estimated probability that FP $= \eta$). Additional details on how to train $\mathcal{F}$ are given Algorithm 4 and Appendix D.2. In the next sections, we will only refer to $\mathcal{F}$ as a general function.

4.3.3 Searching for valid candidate sets

Although our set predictor $\mathcal{F}$ is trained to model either the expected FP or its CDF, it is not necessarily accurate. If $\mathcal{F}$ were simply substituted into Eq. (4.5) or Eq. (4.6), it may not produce valid set predictions. To account for this mismatch, we must carefully calibrate a threshold for accepting candidate sets based on $\mathcal{F}$. At the same time, we also must efficiently search the combinatorial space of candidate sets.

To efficiently calibrate our predictor, we cast our approach into a form of *nested* conformal prediction (Gupta et al., 2019). First, we greedily identify a sequence of nested candidate sets, $\varnothing \subset \mathcal{S}_1 \subset \mathcal{S}_2 \subset \ldots \subset \mathcal{S}_j$, by ranking individual labels $y_c \in \mathcal{Y}$ according to some auxiliary model, and including them one by one into the growing output set $\mathcal{S}_{j+1}$. Notice that, by construction, the number of false positives contained in $\mathcal{S}_j$ is non-decreasing in index j, i.e., $j \leq j' \implies \text{FP}(z, \mathcal{S}_j) \leq \text{FP}(z, \mathcal{S}_{j'})$.

In practice, we find that ranking individual labels by their estimated marginal likelihoods of being true positives, i.e., $p_\theta(y_c \in Z \mid x)$—the same model used in §4.3.2—performs well and avoids the overhead of training an additional scoring model. Importantly, for further efficiency (elaborated on in Remark 4.3.5) we only consider sets up to a maximum size $B \leq |\mathcal{Y}|$, where B is a hyper-parameter that we can set.

Algorithm 4 Calibration of the set score for k-FP validity (§4.3.3).

Definitions: $\mathcal{D}_{\text{cal}}$ is a calibration set, k is the tolerance, and B is a parameter for considering only the top individually ranked candidates. p_θ and $\mathcal{F}$ are the outputs of Algorithm 3.

```
function CALIBRATE($p_\theta$, $\mathcal{F}$, $\mathcal{D}_{\text{cal}}$, $k$, $B$)
    $\mathcal{T}_{\text{cal}} = \{\}$
    for $(x_i, z_i) \in \mathcal{D}_{\text{cal}}$ do
        # Rank top $B$ candidates by individual likelihood.
        $\{y_{i,\pi(1)}, \ldots, y_{i,\pi(B)}\} \leftarrow \textsc{sort}(\mathcal{Y}, p_\theta(y_c \in Z_i \mid x_i))_{1:B}$
        # Construct nested sets using this ordering.
        $\{\mathcal{S}_{i,1}, \ldots, \mathcal{S}_{i,B}\} \leftarrow \{y_{i,\pi(1:j)} \colon j \in \{1, \ldots, B\}\}$
        # Compute nonconformity scores using $\mathcal{F}$.
        $\{v_{i,1}, \ldots, v_{i,B}\} \leftarrow \{\mathcal{F}(x_i, \mathcal{S}_{i,1}), \ldots \mathcal{F}(x_i, \mathcal{S}_{i,B})\}$
        # Cache dependent variables for $\mathrm{FP}_{\max}(x_i, z_i, t)$.
        $\mathrm{FP}_{\max}(x_i, z_i, t) \leftarrow \textsc{cache}(x_i, z_i, v_{i,1:B}, \mathcal{S}_{i,1:B})$
        # Append cached $\mathrm{FP}_{\max}(x_i, z_i, t)$ to the calibration set.
        $\mathcal{T}_{\text{cal}} \leftarrow \mathcal{T}_{\text{cal}} \cup \{\mathrm{FP}_{\max}(x_i, z_i, t)\}$
    # Use Eq. (4.12) to find a $k$-FP valid set score threshold.
    $t_k \leftarrow \textsc{find\_threshold}(\mathcal{T}_{\text{cal}}, B, k)$
    return $t_k$
```

Next, we compute a set nonconformity score v_j (assumed to be finite) for each candidate set $\mathcal{S}_j$ using $\mathcal{F}$, where $v_j := \mathcal{F}(x, \mathcal{S}_j)$. Finally, we define the worst-case number of false positives over all nested candidate sets $\mathcal{S}_{1:B}$ having nonconformity scores less than t (given the input x with label set z) as

$$\mathrm{FP}_{\max}(x, z, t) := \max\Big\{\mathrm{FP}(z, \mathcal{S}_j) \colon v_j < t\Big\}. \tag{4.10}$$

If this set is empty, then $\mathrm{FP}_{\max}$ is 0. Due to our nested construction, this is also simply the number of false positives contained in the largest set $\mathcal{S}_j$ satisfying $v_j < t$. It is simple to show that $\mathrm{FP}_{\max}$ is non-decreasing in t, as stated below.

Lemma 4.3.3 (Monotonicity)**.** *For sets $\mathcal{S}_j$ and scores v_j and $\mathrm{FP}_{\max}(x, z, t)$, we have*

$$t \leq t' \implies \mathrm{FP}_{\max}(x, z, t) \leq \mathrm{FP}_{\max}(x, z, t'). \tag{4.11}$$

Proof. See Appendix D.1.1. □

Using this property, we can find a *maximal* threshold t to use as a "cutoff point" for the sequence of nested candidate sets such that $\mathrm{FP}_{\max}$ is controlled.

Theorem 4.3.4 (FP-CP). *Assume that examples $(X_i, Z_i) \in \mathcal{X} \times 2^{\mathcal{Y}}$, $i = 1, \ldots, n+1$ are exchangeable. For each example i, let $S_{i,j}$, $j = 1, \ldots, B$ (where $B \leq |\mathcal{Y}|$ is a finite hyper-parameter) be candidate sets, where finite random variable $V_{i,j} = \mathcal{F}(X_i, \mathcal{S}_{i,j})$ is a set nonconformity score. For tolerances $k \in \mathbb{R}_{>0}$ and $\delta \in (0, 1)$ define random variables T_k and $T_{k,\delta}$ (based on the first n examples) as*

$$T_k := \sup\left\{t \in \mathbb{R}\colon \frac{B + \sum_{i=1}^{n} \mathrm{FP}_{\max}(X_i, Z_i, t)}{n+1} \leq k\right\} \quad \textit{and} \tag{4.12}$$

$$T_{k,\delta} := \sup\left\{t \in \mathbb{R}\colon \frac{\sum_{i=1}^{n} \mathbf{1}\{\mathrm{FP}_{\max}(X_i, Z_i, t) \leq k\}}{n+1} \geq 1 - \delta\right\}, \tag{4.13}$$

where $\mathrm{FP}_{\max}$ *is as defined in Eq. (4.10). Then we have that*

$$\mathbb{E}\Big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\Big] \leq k, \quad \textit{and} \tag{4.14}$$

$$\mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \leq k\Big) \geq 1 - \delta. \tag{4.15}$$

Proof. See Appendix D.1.2. □

Remark 4.3.5. The hyper-parameter B plays an important role when controlling for k-FP. T_k may be very conservative if $B = |\mathcal{Y}|$ and $|\mathcal{Y}|$ is very large, to the point where $T_k = -\infty$ always if $|\mathcal{Y}| > k(n+1)$. It can therefore be beneficial to truncate the considered label space $\mathcal{Y}$ for an example x to only the top $B \ll k(n+1)$ individual candidates, $\{y_1, \ldots, y_B\} \in \mathcal{Y}^B$. For example, for text generation tasks (like machine translation), $\mathcal{Y}$ is infinite, but we can restrict our predictions to a subset of the top B beam search candidates (where B can still be reasonably large). Nevertheless, this does not come for free: a smaller B may result in fewer true positives.

Remark 4.3.6. No constraints are placed on the underlying set function $\mathcal{F}$; i.e., it need not be a DeepSets architecture. If, however, $\mathcal{F}$ is a good estimator of $\mathrm{FP}(Z, \mathcal{S}) \mid X$, then our method is more likely to identify sets that are approximately valid condi-

Algorithm 5 Set prediction using calibrated t_k (§4.3.4).

Definitions: $x_{\mathrm{n+1}}$ is a test point, B is the same input as for Algorithm 4, and t_k is the output of Algorithm 4. p_θ and $\mathcal{F}$ are the outputs of Algorithm 3.

```
function PREDICT($x_{n+1}$, $p_\theta$, $\mathcal{F}$, $t_k$, $B$)
    # Repeat lines 12-14 to compute $\mathcal{S}_{n+1,1:B}$ and $v_{n+1,1:B}$.
    # Identify indices of candidate sets that pass threshold $t_k$.
    $\mathcal{J} \leftarrow \{j \in \{1, \ldots, B\} \colon v_{n+1,j} < t_k\}$
    # Choose the largest sized set among filtered candidates.
    $\mathcal{C}_k(x_{n+1}) \leftarrow \mathcal{S}_{n+1,\max \mathcal{J}}$
    return $\mathcal{C}_k(x_{n+1})$
```

tioned on $X_{n+1} = x_{n+1}$, which we investigate empirically in §5.5.

Remark 4.3.7. It is useful to note that nestedness of $\mathcal{S}_{i,j}$ is not necessary for the above calibration to hold (it is used for efficiency). Monotonicity of $\mathrm{FP}_{\max}$ is sufficient.

An example implementation of this calibration procedure is given in Algorithm 4.

4.3.4 Selecting the final output set

The main consequence of Theorem 4.3.4 is that, using the calibrated nonconformity threshold T_k or $T_{k,\delta} = t^*$, we can construct a collection of sets that are simultaneously valid by keeping all candidate sets with scores less than t^*. Specifically, we are free to select *any* set in the filtered set of candidates $S_{n+1,j}$, $j \in \mathcal{J}$ where $\mathcal{J} := \{j \colon v_{n+1,j} < t^*\}$, as a valid output. Ideally, we would be able to follow the the oracle strategy in returning the smallest set with the highest number of true positives. This would make our predictions *efficient*, in the sense that we are not including more false positives than necessary (even if the total is still $\leq k$). A reasonable choice is to then choose $\mathcal{S}_{j^*}$ where $j^* := \arg\max_{j \in \mathcal{J}} |\mathcal{S}_j| - \mathcal{F}(x, \mathcal{S}_j)$; but this can be sub-optimal if $\mathcal{F}$ is not accurate. As a greedy, but effective, approach we simply take the *largest* set as our final output, which has maximal TPR. We formalize this in Proposition 4.3.8, and provide its implementation in Algorithm 5.

Proposition 4.3.8 (Greedy FP-CP)**.** *Let $T_\circ$ denote either T_k or $T_{k,\delta}$. Then random candidate sets $\mathcal{S}_{n+1,j}$, $\forall j \in \mathcal{J} := \{j \colon V_{n+1,j} < T_\circ\}$, are valid. Furthermore, among* *indices $\mathcal{J}$, $\max \mathcal{J}$ indexes the nested set with the highest TPR.*

Proof. See Appendix D.1.3. □

We discuss some additional extensions, and limitations, of our method in Appendix D.5.

4.4 Experimental setup

In this section, we outline our tasks and models. We also describe our evaluation and baselines. For all experiments, we set B to 100. Table 4.1 provides statistics for the datasets used in experiments. Appendix D.3 contains additional details.

4.4.1 Tasks

In-silico screening for drug discovery. The goal of in-silico screening is to identify potentially effective drugs to manufacture and test. We use the ChEMBL database (Mayr et al., 2018) to screen molecules for combinatorial constraint satisfaction, where given a constraint such as "*has property A but not property B*," we want to identify the subset of molecules from a given set of candidates that have the desired attributes. We partition the dataset both by molecules and property combinations, so that at test time the model makes predictions on combinations it has never been tested on before (after being trained on the same properties, but seen in different combinations), over a pool of molecules that it has never seen before. Scores for candidate molecules are obtained via an ensemble of directed MPNNs using `chemprop` (Yang et al., 2019).

Object detection. We consider the task of placing bounding boxes around all objects of a certain type (such as a person) that are present in an image (of which there may be few, many, or none). We use the MS-COCO dataset (Lin et al., 2014), a dataset with images of everyday scenes containing 80 object types (e.g., person, bicycle, dog, car, etc). We extract typed bounding box candidates (i.e., tuples of both location *and* category) using an EfficientDet model (Tan et al., 2020) with non-maximum

Dataset	Input	# Examples	# Negatives	# Positives	% Empty
In-silico screening	SMILES	5,000	85 (50-97)	15 (3-50)	0.0
Object detection	Image	3,000	96 (89-98)	4 (2-11)	1.1
Entity extraction	Text	3,453	99 (97-100)	1 (0-3)	20.2

Table 4.1: Dataset statistics (test split). Numbers are reported with respect to the top $B = 100$ candidates per example. The median number of positives and negatives per example is given, in addition to their 16th-84th percentiles. We also report the percentage of examples that have "empty" label sets with no positives (i.e., $|z| = 0$).

suppression. True positives are defined as boxes that have an intersection over union (IoU) > 0.5 with a matching annotation of the same type.

Entity extraction. In entity extraction, we are interested in identifying all named entities that appear in a tokenized sentence x of length l, where $x = \{w_1, \ldots, w_l\}$, and classifying them into appropriate categories. A named entity is a proper noun, demarcated by a contiguous span $\{w_{\text{start}}, \ldots, w_{\text{end}}\} \subseteq x$ of the input sentence, that can be associated with a particular class of interest (such as a person, location, organization, or product). We report results on the CoNLL NER dataset (Tjong Kim Sang and De Meulder, 2003), where we use the PURE span-based entity extraction model of Zhong and Chen (2021) to individually score all $\mathcal{O}(l^2)$ candidate spans. We consider exact span predictions of the correct category to be true positives, and all others to be false positives. Many sentences contain no entities, and have no true positives.

4.4.2 Evaluation

For each task we learn all models on a training set, perform model selection on a validation set, and report final results as the average over 1000 random trials on a test set, where in each trial we partition the data into 80% calibration ($x_{1:n}$) and 20% prediction points (x_{n+1}). To compare across k, we plot each metric as a function of k (up to $k = B$), and compute the area under the curve (AUC). Shaded regions show the 16-84th percentiles across trials. In addition to TPR (our main metric), as our method already guarantees marginal FP-validity, we also compute the size-stratified k-FP (SSFP$_k$) and (k, δ)-FP ($SSFP_{k,\delta}$) violation (Angelopoulos et al., 2021b), see

Appendix D.3.1. Lower size-stratified violation suggests that a classifier has better conditional coverage. We also report average FP results in Table 4.2.

4.4.3 Baselines

We compare our FP-CP (NN) method using DeepSets to the following baselines:

1. **Top-k.** We naïvely take the top k' fixed predictions for any x_{n+1}, where k' is found using average performance on the calibration set (without any correction factors, so it is *not* guaranteed to be valid). Note that this k' can be (and mostly is) different than the user-specified k for FP (i.e., it will be higher).

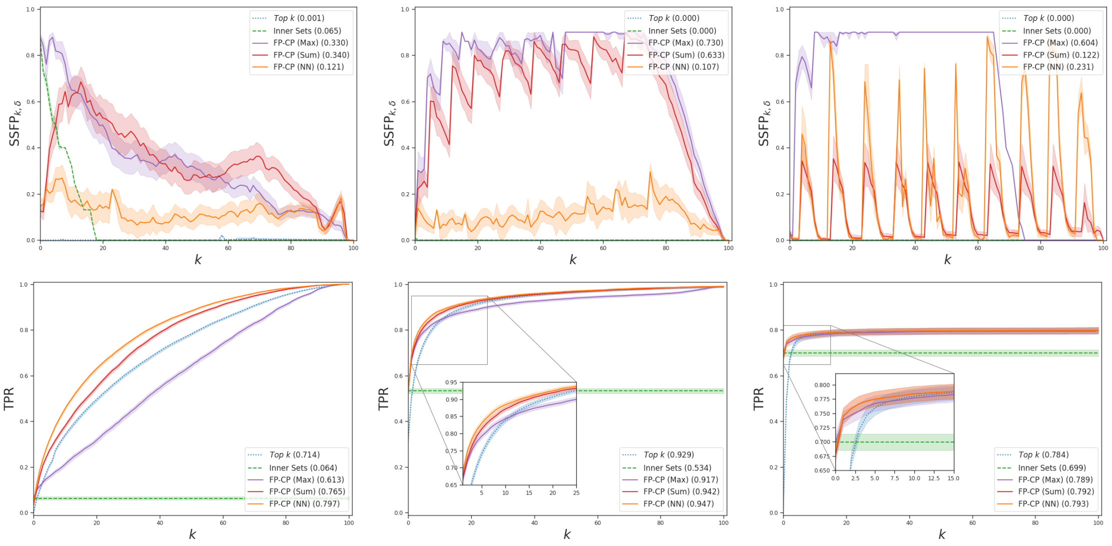
2. **Outer Sets @ 90.** We use the (one-sided) multi-label conformal prediction technique of Cauchois et al. (2021) to bound $\mathbb{P}(Z_{n+1} \subseteq \mathcal{C}_\epsilon(X_{n+1})) \geq 0.90$. Though not directly comparable, we use this to benchmark our method against sets that preserve marginal coverage (at a typical level). For simplicity, we use the direct inner/outer method without dynamic CQC quantiles (which we found to be similar).

3. **Inner Sets.** Again, we use the (one-sided) method of Cauchois et al. (2021), this time to bound $\mathbb{P}(\mathcal{C}_\epsilon(X_{n+1}) \subseteq Z_{n+1}) \geq 1 - \epsilon$ at level $\epsilon = k/B$ (recall that $B \leq |\mathcal{Y}|$ is the truncation parameter, and the FP upper bound) for k-FP control and at level $\epsilon = \delta$ for (k, δ)-FP control. It is straightforward to show that these levels of ϵ conservatively achieve k-FP and (k, δ)-FP control.

4. **Independent scoring (max).** We take $\mathcal{F}(x, \mathcal{S})$ to be the maximum individual label uncertainty in $\mathcal{S}$, $\max\{1 - p_\theta(y_c \in Z \mid x) \colon y_c \in \mathcal{S}\}$. This is equivalent to choosing labels independently. Calibration uses the same FP-CP algorithm (the maximum score functions as a drop-in replacement for the NN-based set score).

5. **Cumulative scoring (sum).** We take $\mathcal{F}(x, \mathcal{S})$ to be the cumulative individual label uncertainty in $\mathcal{S}$, $\sum_{y_c \in \mathcal{S}} 1 - p_\theta(y_c \in Z \mid x)$. We calibrate p_θ using Platt scaling (Platt, 1999). Calibration uses the same FP-CP algorithm.

Baseline (1) contrasts our approach with what is normally a "first thought" in prac-



(a) In-silico screening (b) Object detection (c) Entity extraction

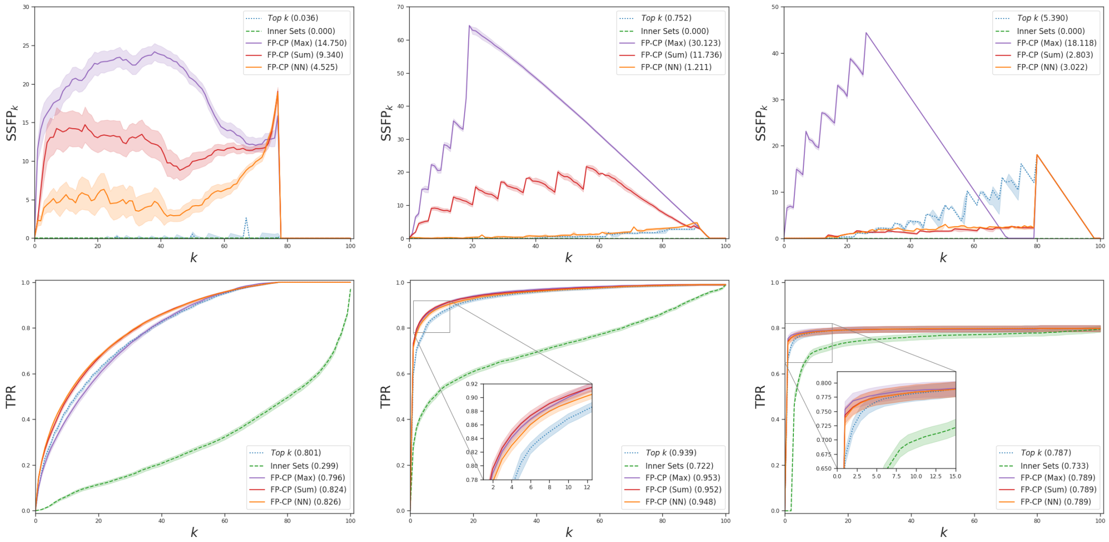
Figure 4-2: (k, δ)-FP results as a function of k up to $k = B = 100$ ($\delta = 0.1$). The top row plots SSFP$_{k,\delta}$ violation (lower is better). The bottom row plots TPR (higher is better). Compared to the other baselines, the DeepSets approach (NN) has the best (or close to) TPR AUC, while having the lowest (or close to) SSFP$_{k,\delta}$ violation.

tice, (2) and (3) test the efficacy of our system over existing techniques, and (4) and (5) demonstrate our FP-CP calibration with simpler alternatives for $\mathcal{F}$.

4.5 Experimental results

We now present our empirical results. Figure 4-3 and Figure 4-2 present AUC results, computed over all values of ϵ, for all tasks. Table 4.2 reports additional absolute results for a number of reference k values, focusing on the in-silico screening task. Appendix D.4 contains additional discussion.

Limiting false positives. The top rows of Figures 4-2 and 4-3 show the size-stratified violation for (k, δ)-FP and k-FP, respectively. Across values of k, FP-CP (NN) typically achieves substantially *lower* worst-case violations than either max or sum scoring alternatives, (though, in some cases, the magnitude of SSFP can depend strongly on k). The Top-k and Inner Sets approaches also prevent large violations (though, by itself, this result is not necessarily impressive, as always returning an empty set will



(a) In-silico screening (b) Object detection (c) Entity extraction

Figure 4-3: k-FP results as a function of k up to $k = B = 100$. The top row plots SSFP$_k$ violation (lower is better). The bottom row plots TPR (higher is better). As k grows, our methods achieve high TPR. Consistent with Figure 4-2, the DeepSets (NN) approach demonstrates high TPR and low SSFP$_k$ across tasks.

lead to SSFP $= 0$). When accounting for TPR (bottom rows), we see that our FP-CP methods demonstrate stronger performance.

Maximizing true positive rates. The bottom rows of Figures 4-2 and 4-3 plot TPR and AUC across k, while Table 4.2 details results for several representative individual configurations. On the screening task, we see that our FP-CP (NN) method provides significantly higher TPR than other baselines. For example, allowing no more than 5 false positives leads to a TPR of 36.1% with k-FP. In comparison, the TPR of Top-k is only 29.8%. As might be expected, the advantage of the DeepSets approach underlying FP-CP (NN) over simpler FP-CP scoring mechansims is more pronounced for tasks with higher cardinality label sets, such as in-silico screening versus object detection of entity extraction (see a comparison of dataset characteristics in Table 4.1). Furthermore, since entity extraction contains a high proportion of examples with "empty" label sets, we can see that its TPR asymptotes at the natural rate of answerable examples. Nevertheless, in general, all FP-CP methods (with max, sum, or NN scoring) provide high TPR (exceeding non FP-CP methods) even at low values of k.

	Top k		Inner Sets		FP-CP (Max)		FP-CP (Sum)		FP-CP (NN)	
k-FP:	$\mathbb{E}$[FP]	*TPR*	$\mathbb{E}$[FP]	*TPR*	$\mathbb{E}$[FP]	*TPR*	$\mathbb{E}$[FP]	*TPR*	$\mathbb{E}$[FP]	*TPR*
$k = 5$	4.59	29.8	0.14	2.5	4.98	27.5	4.99	34.1	4.98	36.1
$k = 15$	14.47	53.4	0.88	9.5	14.98	50.7	14.99	58.8	14.99	59.9
$k = 25$	24.51	68.0	1.49	13.4	24.98	66.8	24.99	73.1	24.99	73.2
$k = 35$	34.54	78.2	2.45	18.4	34.97	78.4	34.99	82.6	34.99	82.5
(k, δ)*-FP with* $1 - \delta = 0.9$:										
	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*	FP $\leq k$	*TPR*
$k = 5$	100.0	20.5	96.6	6.36	90.0	15.8	90.0	27.2	90.0	31.6
$k = 15$	94.7	42.4	99.5	6.36	90.0	26.7	90.0	47.4	90.0	55.3
$k = 25$	96.6	55.7	100.0	6.36	90.0	37.4	90.0	62.3	90.0	69.0
$k = 35$	97.5	66.2	100.0	6.36	90.0	49.1	90.0	74.0	90.0	79.0

Table 4.2: Results for the in-silico screening task on the ChEMBL dataset. For k-FP validity, we report the empirical average of false positives in the prediction sets. For (k, δ)-FP validity we report the percentage of prediction sets with $\leq k$ false positives. TPR is expressed as a percent. Our FP-CP methods meet our target thresholds; using the Inner Sets approach does too, but is conservative (as expected). Applying FP-CP calibration with our DeepSets model (NN) yields substantially higher TPR across various tolerance levels compared to the other baseline scoring mechanisms.

Task	TPR	Avg. FP	Avg. Size
In-silico screening	97.2	63.6	86.6
Object detection	96.1	32.4	38.2
Entity extraction	75.0	0.77	2.31

Table 4.3: Outer Sets applied at coverage 0.90 for comparison. Note that as some examples do *not* have any positives, full coverage in the typical sense is not always achievable. Average FP and set size are reported as absolute values.

Comparison to conformal coverage methods. Table 4.3 gives the results of the Outer Sets method at level 0.90 (a typical tolerance). Indeed, we achieve strong TPR (97.2% for the in-silico screening task), but also incur a high false positive cost in the process (63.6 average FP for in-silico screening). In contrast, our method allows us to directly limit false positives, without losing high TPR empirically (e.g., equivalently controlling for ≤ 63.6 FP, we acheive 97.0% TPR on the in-silico screening task).

4.6 Conclusion

Conformal prediction, in its standard formulation, already grants theoretical performance guarantees that can be critical in many applications. However, as also shown in Chapter 3, naïve applications of CP can yield disappointing results. Even if the target coverage is upheld, the predicted sets may be too large, and too noisy, to be practical. In this chapter, we proposed a method for trading coverage guarantees in favor of strict limits on the number of false positives contained in our prediction sets. Our results show that our method yields classifiers that (1) still achieve strong true positive rates compared to their coverage-seeking counterparts, and (2) predict meaningful output sets with effectively controlled numbers of false positives.

5

Few-shot Conformal Prediction

In this chapter, we develop a new approach to conformal prediction when the target task has limited data available for training. Conformal prediction identifies a small set of promising output candidates with guarantees that the set contains the correct answer with high probability. When training data is limited, however, the predicted set can easily become unusably large. We obtain substantially tighter prediction sets while maintaining desirable marginal guarantees by casting conformal prediction as a meta-learning paradigm over exchangeable collections of auxiliary tasks. Our conformalization algorithm is simple, fast, and agnostic to the choice of underlying model, learning algorithm, or dataset. We demonstrate the effectiveness of this approach across a number of few-shot classification and regression tasks in natural language processing, computer vision, and computational chemistry for drug discovery.

5.1 Introduction

Accurate estimates of uncertainty are important for difficult or sensitive prediction problems that have variable accuracy (Amodei et al., 2016; Jiang et al., 2012, 2018; Angelopoulos et al., 2021b). Few-shot learning problems, in which training data for the target task is severely limited, pose a discouragingly compounded challenge: in general, not only is (1) making accurate predictions with little data hard, but also (2) rigorously quantifying the uncertainty in these few-shot predictions is even harder.

We are interested in creating confident *prediction sets* that provably contain the correct answer with high probability (e.g., 95%), while only relying on a few in-task examples. We continue our focus on conformal prediction (Vovk et al., 2005), where given n examples $(X_j, Y_j) \in \mathcal{X} \times \mathcal{Y}$, $j = 1, \ldots, n$ as training data, that have been drawn exchangeably from some underlying distribution P we aim to construct a set-valued output, $\mathcal{C}_\epsilon(X_{n+1})$, for a new exchangeable test point, $X_{n+1})$, that contains Y_{n+1} with *distribution-free marginal coverage* at a significance level $\epsilon \in (0, 1)$, i.e.,

$$\mathbb{P}\left(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})\right) \geq 1 - \epsilon. \tag{5.1}$$

A conformal model is considered to be *valid* if the frequency of error, $Y_{n+1} \notin \mathcal{C}_\epsilon(X_{n+1})$, does not exceed ϵ. The challenge for few-shot learning, however, is that as $n \to 0$, standard CP methods quickly result in outputs $\mathcal{C}_\epsilon(X_{n+1})$ so large that they lose all utility (e.g., a trivially valid classifier that returns all of $\mathcal{Y}$). A conformal model is only considered to be *efficient* if $\mathbb{E}[|\mathcal{C}_\epsilon(X_{n+1})|]$ is relatively small.

In this chapter we approach this frustrating data sparsity issue by casting conformal prediction as a meta-learning paradigm over exchangeable collections of tasks. By being exposed to a set of similar, auxiliary tasks, our model can *learn to learn quickly* on the target task at hand. As a result, we can increase the data efficiency of our procedure, and are able to produce more precise—and confident—outputs.

Specifically, we use the auxiliary tasks to meta-learn both a *few-shot model* and a *quantile predictor*. The few-shot model provides relevance scores (i.e., *nonconformity scores*, see §1.3) for each possible label candidate $y \in \mathcal{Y}$, and the quantile predictor provides a threshold rule for including the candidate y in the prediction set, $\mathcal{C}_\epsilon(X_{n+1})$, or not. A good few-shot model should provide scores that clearly separate *correct* labels from *incorrect* labels—much like a maximum-margin model. Meanwhile, a good quantile predictor—which is intrinsically linked to the specific few-shot model used—should quantify what few-shot scores correspond to relatively "high" or relatively

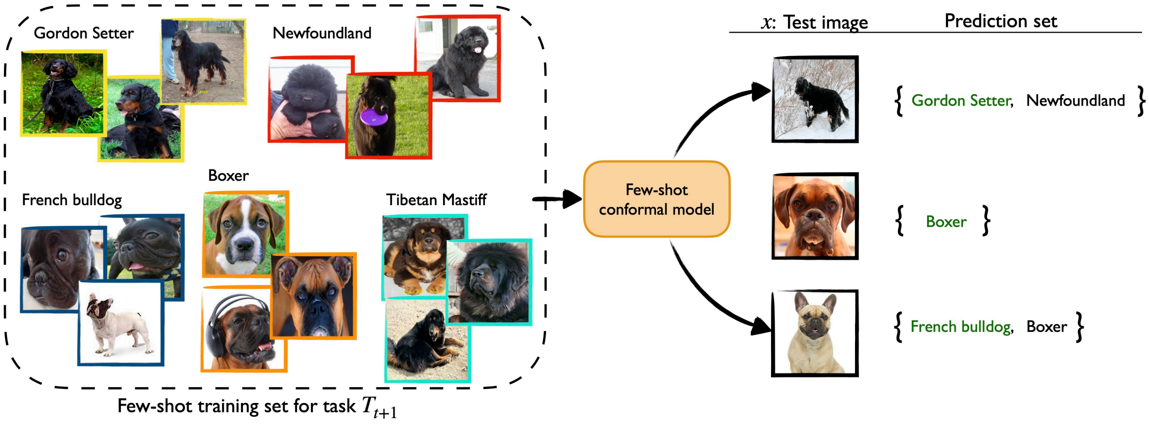


Figure 5-1: A demonstration of our conformalized few-shot learning procedure. Given a base model (e.g., a prototypical network for classification tasks (Snell et al., 2017)) and a few demonstrations of a new task, our method produces a prediction *set* that provably contains the correct answer with some desired probability. Like other meta-learning algorithms, our approach leverages information gained from t other, similar tasks—here to make more precise and confident predictions on the new task, T_{t+1}.

"low" values for that task (i.e., as the name suggests, they infer the target quantile of the expected distribution of few-shot scores). Both of these models must be able to operate effectively given only a few examples from the target task, hence how they are meta-learned over auxiliary tasks becomes crucial.

Consider the example of image classification for novel categories (see Figure 5-1 for an illustration). The goal is to predict the class of a new test image out of several never-before-seen categories—while only given a handful of training examples per category. In terms of auxiliary tasks, we are given access to similarly-framed image classification tasks (e.g., *cat* classes instead of *dog* classes as in Figure 5-1). In this case, we can compute relevance by using a prototypical network (Snell et al., 2017) to measure the Euclidean distance between the test image's representation and the average representation of the considered candidate class's support images (i.e., prototype). Our quantile predictor then computes a "distance cut-off" that represents the largest distance between a label prototype and the test example that just covers the desired percentage of correct labels. Informally, on the auxiliary tasks, the prototypical network will learn efficient features, while the quantile predictor will learn what typically constitutes expected prototypical distances for correct labels when using the trained network.

We demonstrate that these two meta-learned components combine to make an efficient and simple-yet-effective approach to few-shot conformal prediction, all while retaining desirable theoretical performance guarantees. We empirically validate our approach on image classification, relation classification for textual entities, and chemical property prediction for drug discovery.

Contributions. In summary, the main results of this chapter are as follows:

- An extension of conformal prediction to few-shot prediction with auxiliary tasks;
- A meta-learning framework for building set-valued classifiers for new target tasks;
- A empirical demonstration of the utility of our method across classification and regression tasks in NLP, computer vision, and computational chemistry.

5.2 Related work

Conformal prediction. As introduced in §1.3, conformal prediction (Vovk et al., 2005) provides a model-agnostic and finite-sample, distribution-free method for obtaining prediction sets with marginal coverage guarantees. Most pertinent to our work, Linusson et al. (2014) carefully analyze the effects of calibration set size on CP performance. For precise prediction sets, they recommend using at least a few hundred examples for calibration—much larger than the few-shot settings considered here. When the amount of available data is severely restricted, the predicted sets typically become unusably large. Johansson et al. (2015) and Carlsson et al. (2015) introduce similarly motivated approximations to CP with small calibration sets via interpolating calibration instances or using modified p-value definitions. These methods are heuristics, however, and fail to provide finite-sample guarantees. Our work also complements several recent directions that explore conformal prediction in the context of various validity definitions, such as conditional, risk-controlling, admissible, or equalized coverage (Chernozhukov et al., 2019; Cauchois et al., 2021; Kivaranovic et al., 2020; Romano et al., 2019b, 2020a; Bates et al., 2020; Fisch et al., 2021a).

Few-shot learning. Despite the many successes of machine learning models, learning from limited data is still a significant challenge (Bottou and Bousquet, 2008; Lake et al., 2015; Wang et al., 2020b). Our work builds upon the extensive few-shot learning literature by introducing a principled way of obtaining confidence intervals via meta-learning. Meta-learning has become a popular approach to transferring knowledge gained from auxiliary tasks—e.g., via featurizations or computed statistics (Edwards and Storkey, 2017)—to a target task that is otherwise resource-limited (Vinyals et al., 2016; Finn et al., 2017; Snell et al., 2017; Bertinetto et al., 2019; Bao et al., 2020). We leverage the developments in this area for our models (see Appendix E.2.1).

5.3 Few-shot meta conformal prediction

We now propose a general meta-learning paradigm for training efficient conformal predictors, while relying only on a very limited number of *in-task* examples.

At a high level, like other meta-learning algorithms, our approach leverages information gained from t similar tasks in order to perform better on task $t+1$. In our setting we achieve this by learning a more statistically powerful nonconformity measure and quantile estimator than would otherwise be possible using only the limited data available for the target task. Our method uses the following recipe:

1. **Meta-learning.** We meta-learn (and calibrate) a nonconformity measure and quantile predictor over a set of auxiliary tasks;

2. **Target task adaptation.** We adapt our meta nonconformity measure and quantile predictor using the examples we have for our target task;

3. **Prediction.** We compute a prediction set for an input $x \in \mathcal{X}$ by including all labels $y \in \mathcal{Y}$ whose meta nonconformity score is below the predicted $1-\epsilon$ quantile.

Our procedure applied to classification is sketched in Algorithm 6; regression is similar. Our framework is model agnostic, in that it allows for practically any meta-learning implementation for both nonconformity and quantile prediction models.

Algorithm 6 Meta conformal prediction with auxiliary tasks.

Definitions: $T_{1:t+1}$ are exchangeable tasks. $\mathcal{I}_{\text{train}} \cup \mathcal{I}_{\text{cal}}$ are the t tasks used for meta-training and meta-calibration. $z_{1:k} \in (\mathcal{X} \times \mathcal{Y})^k$ are the k support examples for target task T_{t+1}. $x \in \mathcal{X}$ is the target task input. $\mathcal{Y}$ is the label space. ϵ is the significance.

```
function PREDICT($x$, $z_{1:k}$, $T_{1:t}$, $\epsilon$)
    # Learn $\mathcal{N}$ and $\mathcal{P}$ on meta-training tasks (§5.3.2).
    # $\mathcal{N}$ and $\mathcal{P}$ are meta nonconformity/quantile models.
    $\mathcal{N}, \mathcal{P}_{1-\epsilon} \leftarrow \texttt{train}(T_i, i \in \mathcal{I}_{\text{train}})$
    # Predict the $1 - \epsilon$ quantile.
    $\widehat{Q}_{t+1} \leftarrow \mathcal{P}_{1-\epsilon}(z_{1:k}; \phi_{\text{meta}})$
    # Initialize empty output set.
    $\mathcal{M}_\epsilon \leftarrow \{\}$
    # (Note that for regression tasks, where $|\mathcal{Y}| = \infty$, for certain $\mathcal{N}$ the following
    # simplifies to a closed-form interval, making it tractable—see §1.3.)
    for $y \in \mathcal{Y}$ do
        # Compute the nonconformity score for label $y$.
        $\widehat{V}^{(x,y)}_{t+1,k+1} \leftarrow \mathcal{N}((x, y), z_{1:k}; \theta_{\text{meta}})$
        # Compare to the calibrated quantile (§5.3.3). Keep label $y$ if deemed conformal.
        if $\widehat{V}^{(x,y)}_{t+1,k+1} \leq \widehat{Q}_{t+1} + \Lambda(1 - \epsilon, \mathcal{I}_{\text{cal}})$ then
            $\mathcal{M}_\epsilon \leftarrow \mathcal{M}_\epsilon \cup \{y\}$
    return $\mathcal{M}_\epsilon$
```

In the following sections, we break down our approach in detail. In §5.3.1 we precisely formulate our few-shot learning setup with auxiliary tasks. In §5.3.2 and §5.3.3 we describe our meta-learning and meta-calibration setups, respectively. Finally, in §5.3.4 we discuss further theoretical extensions. For a complete technical description of our modeling choices and training strategy for our experiments, see Appendix E.2.

5.3.1 Task formulation

We assume access to t auxiliary tasks, T_i, $i = 1, \ldots, t$, that we wish to leverage to produce tighter uncertainty sets for predictions on a new task, T_{t+1}. Furthermore, we assume that these $t+1$ tasks are *exchangeable* with respect to some task distribution, $P_{\mathcal{T}}$. Here, we treat $P_{\mathcal{T}}$ as a distribution over random distributions, where each task $T_i \in \mathcal{T}$ defines a task-specific distribution, $P_{XY} \sim P_{\mathcal{T}}$, over examples $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. The randomness is in both the task's relation between X and Y, and the task's data.

For each of the t auxiliary tasks, we do not make any assumptions on the amount of data we have (though, in general, we expect them to be relatively unrestricted). On the new task T_{t+1}, however, we only assume a total of k exchangeable training examples. Our goal is then to develop a task-agnostic uncertainty estimation strategy that generalizes well to new examples from the task's unseen test set, $(X_{t+1}^{\text{test}}, Y_{t+1}^{\text{test}})$.[1] Specifically, we desire finite-sample marginal *task* coverage, as follows:

Definition 5.3.1 (Task validity)**.** *Let $\mathcal{M}_\epsilon$ be a set-valued predictor. $\mathcal{M}_\epsilon$ is considered to be* valid *across tasks if for any task distribution $P_\mathcal{T}$ and $\epsilon \in (0, 1)$, we have*

$$\mathbb{P}\Big(Y_{t+1}^{\text{test}} \in \mathcal{M}_\epsilon\left(X_{t+1}^{\text{test}}\right)\Big) \geq 1 - \epsilon. \tag{5.2}$$

Note that we require the marginal coverage guarantee above to hold on average across tasks—which is nuanced difference from the condition expressed in Eq. (5.1) which holds on average over calibration and test samples within a task.

5.3.2 Meta-learning conformal prediction models

Given our collection of auxiliary tasks, we would like to meta-learn both (1) an effective nonconformity measure that is able to adapt quickly to a new task using only k examples; and (2) a quantile predictor that is able to robustly identify the $1 - \epsilon$ quantile of that same nonconformity measure using the same k examples.

Prior to running our meta-learning algorithm of choice, we split our set of t auxiliary tasks into disjoint sets of training tasks, $\mathcal{I}_{\text{train}}$, and calibration tasks, $\mathcal{I}_{\text{cal}}$, where $|\mathcal{I}_{\text{train}}| + |\mathcal{I}_{\text{cal}}| = t$. See Table 5.1 for an overview of the different splits. We use $\mathcal{I}_{\text{train}}$ to learn our meta nonconformity measures and quantile predictors, which we discuss now. Additional technical details are contained in Appendix E.2.1.

[1] For ease of notation, we write X_{t+1}^{test} to denote the $(k+1)$th example of task T_{t+1}, i.e., the new test point after observing k training points. This is equivalent to test point X_{n+1} from §1.3.

	Task Split	# Tasks	# Examples / Task
Auxiliary	Meta-training	$\lvert\mathcal{I}_{\text{train}}\rvert$	$\gg k$
	Meta-calibration	$\lvert\mathcal{I}_{\text{cal}}\rvert$	$k + m_i$
	Test	1	k

Table 5.1: An overview of the data assumptions for a single test task "episode". We use $|\mathcal{I}_{\text{train}}| + |\mathcal{I}_{\text{cal}}| = t$ total auxiliary tasks to create more precise uncertainty estimates for the $(t+1)$th test task. This is repeated for each test task (§5.4). $m_i \gg k$ is the number of extra examples per calibration task that are used to compute an empirical CDF when finding $\Lambda(\beta; \mathcal{I}_{\text{cal}})$—it may vary per task.

Meta nonconformity measure

Let $\mathcal{N}\left((x, y), \mathcal{D}; \theta_{\text{meta}}\right)$ be a *meta nonconformity measure*, where θ_{meta} are meta parameters learned over the auxiliary tasks in $\mathcal{I}_{\text{train}}$, and $\mathcal{D}$ is a test dataset following notation in §1.3. Since θ_{meta} is fixed after the meta training period, $\mathcal{N}$ preserves exchangeability over new collections of exchangeable tasks (i.e., $\mathcal{I}_{\text{cal}}$) and task examples. Let $Z_{i,j} := (X_{i,j}, Y_{i,j})$, $j = 1, \ldots, k$ be the few-shot training data for a task T_i (here i is the task index, while j is the example index). Following the formulation of "full" CP, given a new test point $x \in \mathcal{X}$ and candidate pairing (x, y), the *meta nonconformity scores* are computed as

$$\begin{aligned} V_{i,j}^{(x,y)} &:= \mathcal{N}(Z_{i,j}, Z_{i,1:k} \cup \{(x, y)\}; \theta_{\text{meta}}), \\ V_{i,k+1}^{(x,y)} &:= \mathcal{N}((x, y), Z_{i,1:k} \cup \{(x, y)\}; \theta_{\text{meta}}). \end{aligned} \tag{5.3}$$

As an example, Figure 5-2 demonstrates how we compute $V_{i,k+1}^{(x,y)}$ using the distances from a meta-learned prototypical network following the setting in Figure 5-1.

Computing all $k+1$ scores $|\mathcal{Y}|$ times is typically tractable due to the few number of examples (e.g., $k \approx 16$) and the underlying properties of the meta-learning algorithm supporting $\mathcal{N}$. For example, prototypical networks only require computing a forward pass. A naive approach to few-shot conformal prediction is to exploit this efficiency, and simply run full CP using all $k+1$ data points. Nevertheless, though a strong baseline, using only $k+1$ points to compute an empirical quantile is still suboptimal.

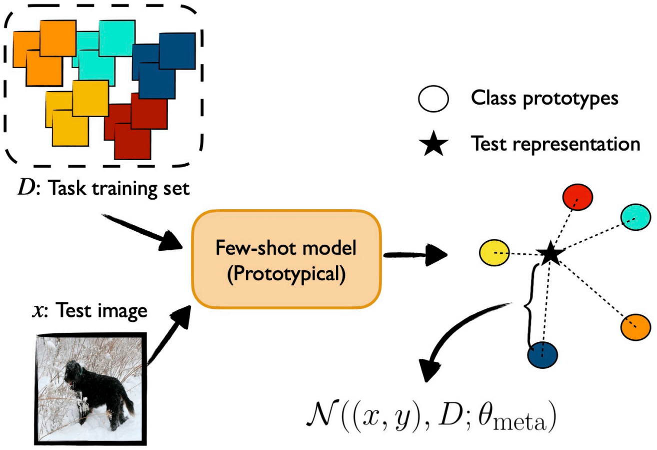

Class prototypes
Test representation
D: Task training set
Few-shot model (Prototypical)
x: Test image
$\mathcal{N}((x, y), D; \theta_{\text{meta}})$

Figure 5-2: An example of using a prototypical network (Snell et al., 2017) to compute meta nonconformity scores. If $\mathcal{N}$ is well-trained, the distance between the test point and the correct class prototype should be small, and the distance to incorrect prototypes large, even when the number of in-task training examples is limited.

As we discuss next, we choose to regress the desired quantile directly from $Z_{i,1:k}$, and disregard the empirical quantile completely. Since we predict the quantile instead of relying on the empirical quantile, we do not have to retain exchangeability for $Z_{i,1:k}$. As a result, we *do not* include (x, y) when calculating $V_{i,j}^{(x,y)}$, as this is faster.

Meta quantile predictor

Let $\mathcal{P}_\beta(\mathcal{D}; \phi_{\text{meta}})$ be a *meta β-quantile predictor*, where ϕ_{meta} are the meta parameters learned over the auxiliary tasks in $\mathcal{I}_{\text{train}}$. $\mathcal{P}_\beta$ is trained to predict the β-quantile of F, where F is the underlying task-specific distribution of nonconformity scores—given $\mathcal{D}$, a dataset of $Z = (X, Y)$ pairs sampled from that task. As some intuition, recall that in calculating Quantile$(\beta; F)$ given exchangeable samples $v_{1:n} \sim F$, we implicitly need to estimate $\mathbb{P}(V_{n+1} \leq v \mid v_{1:n})$. For an appropriate parameterization ψ of F, de Finetti's theorem for exchangeable sequences allows us to write

$$\mathbb{P}(V_{n+1} \leq v \mid v_{1:n}) \propto \int_{-\infty}^{v} \int_{\Psi} p(v \mid \psi) \prod_{i=1}^{n} p(v_i \mid \psi) p(\psi) d\psi dv. \tag{5.4}$$

In this sense, meta-learning over auxiliary task distributions may help us learn a better prior over latent parametrizations ψ—which in turn may help us better model

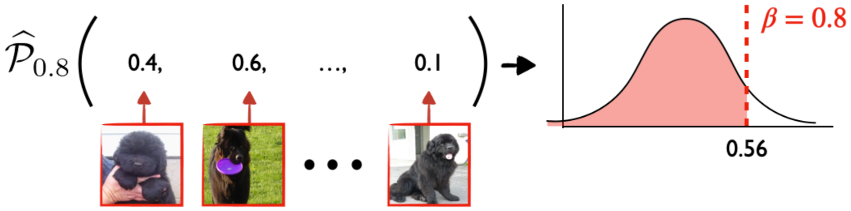


Figure 5-3: An illustration of using our quantile predictor $\mathcal{P}_\beta$ to infer the β-quantile of the distribution of $V^{\text{test}}_{i,k+1}$, given the few examples from T_i's training set. The numbers above each image reflect the leave-one-out scores we use as inputs, see Eq. (5.6).

the β-quantile than we could have, given only k samples and nothing else.

We develop a simple approach to modeling and learning $\mathcal{P}_\beta$. Given the training examples $Z_{i,1:k}$, we use a deep sets model (Zaheer et al., 2017) parameterized by ϕ_{meta} to predict the β-quantile of $V^{\text{test}}_{i,k+1}$, the random variable representing the nonconformity score of the test point, $Z_{i,k+1} := (X_{i,k+1}, Y_{i,k+1})$. We optimize ϕ_{meta} as

$$\phi_{\text{meta}} := \min_{\phi} \sum_{i \in \mathcal{I}_{\text{train}}} \left| \mathcal{P}_\beta\Big(Z_{i,1:k}; \phi\Big) - \text{Quantile}\Big(\beta; V^{\text{test}}_{i,k+1}\Big) \right|^2 \tag{5.5}$$

where we estimate the target, $\text{Quantile}\Big(\beta; V^{\text{test}}_{i,k+1}\Big)$, using $m \gg k$ extra examples sampled from the training task. In practice, we found that choosing to first transform $Z_{i,1:k}$ to *leave-one-out* meta nonconformity scores,

$$L_{i,j} := \mathcal{N}\Big(Z_{i,j}, Z_{i,1:k} \setminus Z_{i,j}; \theta_{\text{meta}}\Big), \tag{5.6}$$

and providing $\mathcal{P}_\beta$ with these scalar leave-one-out scores as inputs, performs reasonably well and is simple to implement.[2] Inference using $\mathcal{P}_\beta$ is illustrated in Figure 5-3.

Training strategy

The meta nonconformity measure $\mathcal{N}$ and meta quantile predictor $\mathcal{P}_\beta$ are tightly coupled, as given a fixed $\mathcal{N}$, $\mathcal{P}_\beta$ learns to model its behavior on new data. A straightforward, but data inefficient, approach to training $\mathcal{N}$ and $\mathcal{P}_\beta$ is to split the collection of

[2] We also experimented with directly encoding $Z_{i,1:k}$, but did not find that it added much benefit.

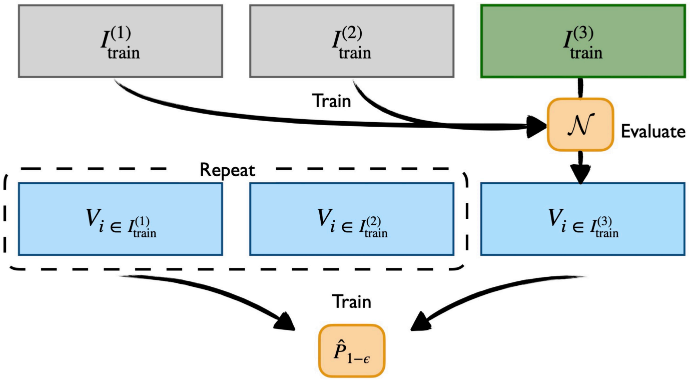


Figure 5-4: An illustration of our strategy for learning meta nonconformity measures, $\mathcal{N}$, and meta quantile predictors, $\mathcal{P}_{1-\epsilon}$. As $\mathcal{P}_{1-\epsilon}$ is trained on the outputs of $\mathcal{N}$, we adopt a cross-fold procedure where we first train $\mathcal{N}$ on a fraction of the data, and evaluate nonconformity scores on the held-out fold. We repeat this process for all k_f folds, and then aggregate them all for training $\mathcal{P}_{1-\epsilon}$.

auxiliary tasks in $\mathcal{I}_{\text{train}}$ in two, i.e., $\mathcal{I}_{\text{train}} = \mathcal{I}^{(1)}_{\text{train}} \cup \mathcal{I}^{(2)}_{\text{train}}$, and then train $\mathcal{N}$ on $\mathcal{I}^{(1)}_{\text{train}}$, followed by training $\mathcal{P}_\beta$ on $\mathcal{N}$'s predictions over $\mathcal{I}^{(2)}_{\text{train}}$. The downside is that both $\mathcal{N}$ and $\mathcal{P}_\beta$ may be sub-optimal, as neither can take advantage of all of $\mathcal{I}_{\text{train}}$.

We employ a slightly more involved, but more data efficient approach, where we split $\mathcal{I}_{\text{train}}$ into k_f folds, i.e., $\mathcal{I}_{\text{train}} = \bigcup_{f=1}^{k_f} \mathcal{I}^{(f)}_{\text{train}}$. We then train k_f separate meta nonconformity measures $\widehat{S}_f$, where we leave out fold f from the training data. Using $\widehat{S}_f$, we compute nonconformity scores on fold f's data, aggregate these nonconformity scores across all k_f folds, and train the meta quantile predictor on this union. Finally, we train another nonconformity measure on *all* of $\mathcal{I}_{\text{train}}$, which we use as our ultimate $\widehat{S}$. This way we are able to use all of $\mathcal{I}_{\text{train}}$ for training both $\mathcal{N}$ and $\mathcal{P}_\beta$. This process is illustrated in Figure 5-4. Note that it is not problematic for $\mathcal{P}_\beta$ to be trained on the collection of $\mathcal{N}$ instances trained on $k_f - 1$ folds, but then later used to model one $\mathcal{N}$ trained on *all* the data, since it will be calibrated (next, in §5.3.3).

5.3.3 Calibrating meta-learned conformal prediction

Though $\mathcal{P}_\beta$ may obtain low empirical error after training, it does not have any guarantees out-of-the-box. Given our held-out set of auxiliary tasks $\mathcal{I}_{\text{cal}}$, however, we can quantify the uncertainty in $\mathcal{P}_\beta$ (i.e., how far off it may be from the true quantile), and calibrate it accordingly. The following lemma formalizes our procedure:

Lemma 5.3.2 (Meta calibration)**.** *Let $\widehat{Q}_i$, $i \in \mathcal{I}_{\text{cal}}$ be (exchangeable) meta β-quantile predictions produced by $\mathcal{P}_\beta$ for tasks T_i, $i \in \mathcal{I}_{\text{cal}}$,*

$$\widehat{Q}_i := \mathcal{P}_\beta(Z_{i,1:k}; \phi_{\text{meta}}) \tag{5.7}$$

Let $V_{i,k+1}^{\text{test}}$ be the meta nonconformity score for a new sample from task T_i, where F_i is its distribution function. Define the correction $\widehat{\Lambda}(\beta; \mathcal{I}_{\text{cal}})$ as

$$\widehat{\Lambda}(\beta; \mathcal{I}_{\text{cal}}) := \inf\left\{\lambda \in \mathbb{R} \colon \frac{1}{|\mathcal{I}_{\text{cal}}| + 1} \sum_{i \in \mathcal{I}_{\text{cal}}} F_i\Big(\widehat{Q}_i + \lambda\Big) \geq \beta\right\}. \tag{5.8}$$

We then have that $\mathbb{P}\Big(V_{t+1,k+1}^{\text{test}} \leq \widehat{Q}_{t+1} + \widehat{\Lambda}(\beta; \mathcal{I}_{\text{cal}})\Big) \geq \beta$.

Proof. See Appendix E.1.1 □

It is important to clarify at this point that calculating $\widehat{\Lambda}(\beta; \mathcal{I}_{\text{cal}})$ requires knowledge of the true meta nonconformity distribution functions, F_i, for all calibration tasks. For simplicity, we write Lemma 5.3.2 and the following Theorem 5.3.3 as if these distribution functions are indeed known (again, only for calibration tasks). In practice, however, we typically only have access to an *empirical* distribution function over m_i task samples. In this case, Lemma 5.3.2 holds in expectation over task samples $Z_{i,k:k+m_i}$, as for an empirical distribution function of m points, $\widehat{F}_m$, we have $\mathbb{E}[\widehat{F}_m(v)] = F(v)$. For large enough m_i, concentration results also suggest that we can approximate F_i with little error given a particular sample (this is the focus of §5.3.4).

That said, in a nutshell, Lemma 5.3.2 allows us to adjust for the error in $\mathcal{P}_\beta$, such that it is guaranteed to produce valid β-quantiles on average. We can then perform conformal inference on the target task by comparing each meta nonconformity score for a point $x \in \mathcal{X}$ and candidate label $y \in \mathcal{Y}$ to the calibrated meta quantile, and keep all candidates with nonconformity scores that fall below it.

Theorem 5.3.3 (Meta CP). *Assume that tasks T_i, $i \in \mathcal{I}_{\text{cal}}$ and T_{t+1} are exchangeable with known nonconformity distribution functions F_i, $\forall i \in \mathcal{I}_{\text{cal}}$. For any meta quantile predictor $\mathcal{P}_{1-\epsilon}$, meta nonconformity measure $\mathcal{N}$, and tolerance $\epsilon \in (0, 1)$, define the meta conformal set (based on $\mathcal{I}_{\text{cal}}$ and the k examples of task T_{t+1}) at $x \in \mathcal{X}$ as*

$$\mathcal{M}_\epsilon(x) := \Big\{ y \in \mathcal{Y} \colon V_{t+1,k+1}^{(x,y)} \leq \widehat{Q}_{t+1} + \widehat{\Lambda}(1 - \epsilon; \mathcal{I}_{\text{cal}}) \Big\}, \tag{5.9}$$

where $\widehat{Q}_{t+1}$ is the result of running $\mathcal{P}_{1-\epsilon}$ on the k training examples of task T_{t+1},

$$\widehat{Q}_{t+1} := \mathcal{P}_\beta(Z_{t+1,1:k}; \phi_{\text{meta}}). \tag{5.10}$$

Then $\mathcal{M}_\epsilon(X_{t+1}^{\text{test}})$ satisfies Eq. (5.2).

Proof. See Appendix E.1.2 □

Theorem 5.3.3 guarantees coverage *marginally* over tasks, as specified in Eq. (5.2). Naturally, we might also want to ask how well we can be guaranteed to perform on a *specific* task. This strongly depends on the quality of the quantile predictor $\mathcal{P}_{1-\epsilon}$, as well as how many in-task examples we have for the target task. For example, if the predictor is consistent, its predictions will improve with k.

Definition 5.3.4 (Consistency). *We say $\mathcal{P}_{1-\epsilon}$ is an asymptotically consistent estimator of the $1 - \epsilon$ quantile if $\lim_{k\to\infty} \mathcal{P}_{1-\epsilon}(Z_{i,1:k}; \phi_{\text{meta}}) \xrightarrow{p} \text{Quantile}(1 - \epsilon, F_i)$, where k is the number of in-task examples for task $t_i \in \mathcal{T}$.*

In other words, asymptotically consistent $\mathcal{P}_{1-\epsilon}$ converge in probability to the true quantile given enough in-task data (where the randomness is over the draw of in-task data). Under this assumption, and the assumption that tasks are i.i.d. with continuous distribution functions, we achieve task-conditional coverage asymptotically.

Proposition 5.3.5 (Asymptotic meta CP). *Assume that tasks T_i, $i \in \mathcal{I}_{\text{cal}}$ and T_{t+1} are i.i.d., and that F_i are known and continuous. If $\mathcal{P}_{1-\epsilon}$ is also asymptotically* *consistent, then the meta conformal set $\mathcal{M}_\epsilon$ based on $\mathcal{P}_{1-\epsilon}$ achieves asymptotically valid task-conditional coverage, where*

$$\lim_{k\to\infty} \mathbb{P}\Big(\mathbb{P}\Big(Y^{\text{test}}_{t+1,k+1} \notin \mathcal{M}_\epsilon(X^{\text{test}}_{t+1,k+1}) \mid T_{t+1} = t_{t+1}\Big) \leq \epsilon\Big) = 1 \tag{5.11}$$

Proof. See Appendix E.1.3. □

The outer probability is taken over the $k(|\mathcal{I}_{\text{cal}}| + 1)$ in-task examples sampled for quantile prediction for all tasks, while the inner probability is over the test samples for the target task. Furthermore, Eq. (5.11) bounds the *under-coverage* rate, but not the *over-coverage* rate (i.e., how conservative the predictions are). How conservative the meta conformal predictor is can also depend on the number of calibration tasks. Of course, the underlying assumptions of Proposition 5.3.5 are fairly strong. Yet, at its essence, this result simply claims that as the number of in-task samples k increases, our under-coverage rate becomes vanishingly small on all tasks, not just on average (for appropriate $\mathcal{P}_{1-\epsilon}$ and F_i). Intuitively speaking, as we get more in-task data, we can expect our quantile predictor to be more accurate, and therefore require less adjustment. By itself, this is not particularly inspiring: after all, standard CP also becomes viable as $k \to \infty$. Rather, the takeaway is that this desirable behavior is nicely preserved in our meta setup. In Figure 5-6 we demonstrate that in our experiments $\mathcal{P}_{1-\epsilon}$ indeed becomes significantly more accurate as k grows.

5.3.4 Approximate meta-learned conformal prediction

Recall that a key assumption in the theoretical results established in the previous section is that the distribution functions of our calibration tasks (F_i where $i \in \mathcal{I}_{\text{cal}}$) are *known*. In this section we turn to analyze the practical setting where these F_i must be estimated empirically, i.e.,

$$\widehat{F}_{m_i}(v) := \frac{1}{m_i}\sum_{j=1}^{m_i} \mathbf{1}\{V^{\text{test}}_{i,k+j} \leq v\}, \tag{5.12}$$

where $\widehat{F}_{m_i}(v)$ is computed over an empirical sample of m_i points. In this case, Theorem 5.3.3 holds only in expectation over the m_i samples chosen for each the calibration tasks, as $\mathbb{E}[\widehat{F}_{m_i}(v)] = F_i(v)$. However, standard concentration results also suggest that F_{m_i} can approximate F_i with little error, given enough empirical samples (which, in general, we assume we have for our calibration tasks). We now further adapt Theorem 5.3.3 to be conditionally valid with respect to the specific labeled examples that are used when replacing each task F_i with its plug-in estimate, $\widehat{F}_{m_i}$.

First, we formalize a PAC-type 2-parameter validity definition (similar to *training-conditional* CP in Vovk (2012)):

Definition 5.3.6 ((δ, ϵ) task validity)**.** *$\mathcal{M}_{\delta,\epsilon}$ is* (δ, ϵ) task valid *if for any task distribution $P_{\mathcal{T}}$, $\epsilon \in (0,1)$, and $\delta \in (0,1)$,*

$$\mathbb{P}\Big(\mathbb{P}\Big(Y_{t+1}^{\text{test}} \in \mathcal{M}_{\delta,\epsilon}\left(X_{t+1}^{\text{test}}\right)\Big) \geq 1 - \epsilon\Big) \geq 1 - \delta \tag{5.13}$$

The outer probability is taken over the data samples used to compute F_{m_i} for each calibration task, while the inner probability is still over the choice of calibration and test tasks and their examples (but conditioned on the particular samples chosen to compute F_{m_i} for each task). The basic idea is to include a secondary confidence level δ that allows us to control how robust we are to sampling variance in our estimation of calibration task quantiles when computing $\widehat{\Lambda}(\beta; \mathcal{I}_{\text{cal}})$, our correction factor. In practice, this equates to simply adjusting the tolerance ϵ to be slightly stricter. Proposition 5.3.7 formalizes our sample-conditional approach.

Proposition 5.3.7 (Sample-conditional meta CP)**.** *Assume that all $|\mathcal{I}_{\text{cal}}| = l$ calibration tasks are i.i.d., where for each task we have a fixed i.i.d. dataset. That is, for task T_i, we have drawn m_i i.i.d. training examples, $\left(x_{i,j}, y_{i,j}\right)$, $j = 1, \ldots, m_i$. For any $\delta \in (0,1)$, $\epsilon \in (0,1)$, and $\alpha \in \left(0, 1 - (1-\delta)^{\frac{1}{l}}\right)$, define the adjusted ϵ' as*

$$\epsilon' \leq \epsilon - \sqrt{\frac{-2}{l^2}\left(\sum_{i \in \mathcal{I}_{\text{cal}}} \gamma_i^2\right) \log\left(1 - \frac{1-\delta}{(1-\alpha)^l}\right)} \tag{5.14}$$

where $\gamma_i = \sqrt{\frac{\log(2/\alpha)}{2m_i}}$. *Then* $\mathcal{M}_{\delta,\epsilon'}(X_{t+1}^{\text{test}})$ *satisfies Eq.* (5.13).

Proof. See Appendix E.1.4 □

Remark 5.3.8. This result is still not *calibration set*-conditional, which would allow for a fixed set of *calibration tasks* (versus a marginal guarantee over all tasks). This must account for a second source of sampling variance (the tasks), and can be addressed using techniques similar to the one above (see follow-up in Park et al. (2022)).

Remark 5.3.9. Perhaps most importantly, this result is also still not *test task*-conditional, which would allow the bound to hold conditioned on a particular test task.

Experimentally, in §5.5 we demonstrate that we can achieve (δ, ϵ) task valid meta conformal predictors, while only sacrificing a small amount of efficiency relative to our unadjusted methods, even for low tolerances δ (i.e., $\delta = 0.1$).

5.4 Experimental setup

In this section, we outline the tasks and evaluation procedure we use to empirically validate our approach. Additional technical details, such as data preprocessing, modeling choices, and training procedures, are included in Appendix E.2.2.

5.4.1 Evaluation tasks

Image classification (CV). As introduced in §5.1, the goal of few-shot image classification is to train a computer vision model that generalizes to entirely new image classes at test time. We use the *mini*ImageNet dataset (Vinyals et al., 2016), a downsampled version of a subset of classes from ImageNet (Deng et al., 2009). *mini*ImageNet contains 100 classes that are divided into training, validation, and test class splits.

Within each class partition, we construct K-shot N-way tasks, where K examples per class are used to discriminate between a sample of N distinct, novel classes. We use $K = 16$ and $N = 10$ in our experiments, for a total of $k = 160$ training examples. In order to avoid label imbalanced accuracy, however, we choose to focus on Mondrian CP (Vovk et al., 2005), where validity is guaranteed across class type. In other words, we design our task validity guarantee in Eq. (5.2) to hold conditionally on the test class (e.g., if discriminating among the dog classes in Figure 5-1, we require coverage to hold independently over all test images that are Boxers, Gordon Setters, Newfoundlands, etc). Our meta nonconformity measure consists of a standard prototypical network based on top of a CNN encoder.

Relation classification (NLP). Relation classification focuses on identifying the relationship between two entities mentioned in a given natural language sentence (here, English). An example sentence would be "*Newton* served as the president of *the Royal Society*", where the relation between the entities (*Newton*, *the Royal Society*) can then be inferred as *member_of* (a predetermined category). In few-shot relation classification, the goal is to train an NLP model that generalizes to entirely new sentences and entity relationship types at test time. We use the FewRel 1.0 dataset (Han et al., 2018), which consists of 100 relations derived from $70k$ Wikipedia sentences. Like *mini*ImageNet, the relation types are divided into training, validation, and test splits. We only use training/validation splits (the test set is hidden). Within each partition, we sample K-shot N-way classification episodes (again with $K = 16$ and $N = 10$ and Mondrian CP for validity per relation type, as in our CV task). Our meta nonconformity measure consists of a prototypical network on top of a CNN encoder over GloVe word embeddings (Pennington et al., 2014).

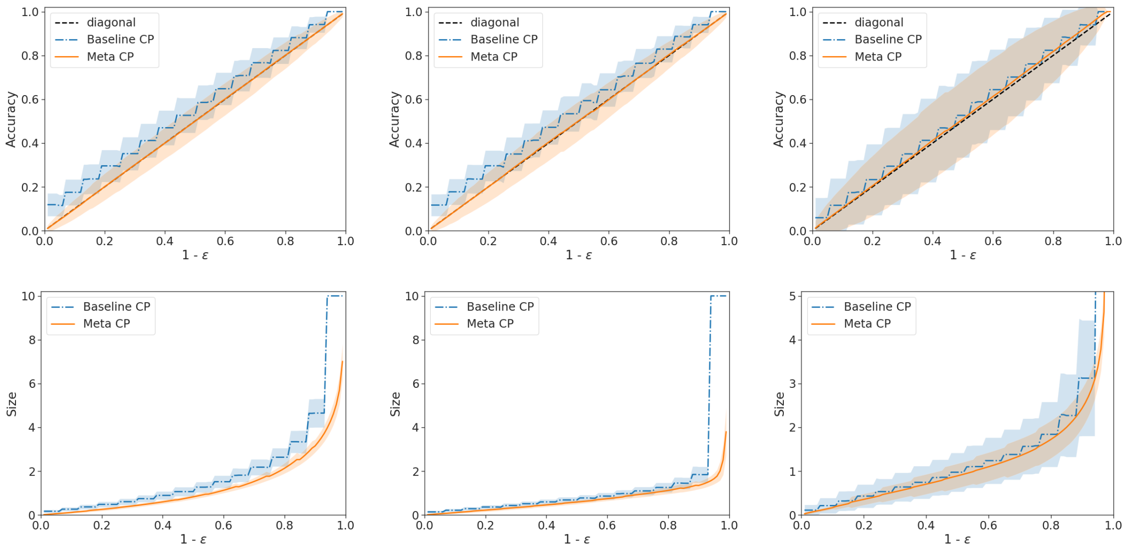
Chemical property prediction (Chem). In-silico screening of chemical compounds is an important task for drug discovery. Given a new molecule, the goal is to predict its activity for a target chemical property. We use the ChEMBL dataset (Mayr et al., 2018), a manually curated database of bioactive molecules with drug-like properties, and regress the pChEMBL value (a normalized log-activity metric) for individual molecule-property pairs. We select a subset of 296 sufficiently diverse assays from ChEMBL, and divide them into training (208), validation (44), and test (44) splits. Within each partition, each assay's pChEMBL values are treated as a real-valued regression task. We use $k = 16$ training samples per task. Our meta nonconformity measure consists of a few-shot, closed-form ridge regressor (Bertinetto et al., 2019) on top of a directed Message Passing Neural Network molecular encoder trained using `chemprop` (Yang et al., 2019). For our full CP regression baseline, we apply ridge regression at test time via the RRCM algorithm (Nouretdinov et al., 2001) on top of the same meta-learned MPNN encoded features.

5.4.2 Evaluation metrics

For each experiment, we use proper training, validation, and test meta-datasets. We use the meta-training tasks to learn all meta nonconformity measures $\mathcal{N}$ and meta quantile predictors $\mathcal{P}$. We perform model selection for CP on the meta-validation tasks, and report final numbers on the meta-test tasks. For all methods, we report marginalized results over 5000 random trials, where in each trial we partition the data into l calibration tasks ($T_{1:l}$) and one target task (T_{t+1}). In all plots, shaded regions show +/- the standard deviation across trials. We use the following metrics:

Prediction accuracy. We measure accuracy as the rate at which the target label $y \in \mathcal{Y}$ is contained within the predicted label set. For classification problems, the prediction is a discrete set, whereas in regression the prediction is a continuous interval. To be valid, a conformal model should have an average accuracy rate $\geq 1 - \epsilon$. *Higher* accuracy rates than necessary, however, are not necessarily *better* (i.e., we can afford to choose the output set more aggressively).

Prediction size (↓). We measure the average size of the output (i.e., $|\mathcal{C}_\epsilon|$) as a proxy for how *precise* the model's predictions are. In the conformal prediction this is commonly referred to as *efficiency* (Vovk et al., 2016). The goal is to make the prediction set as small as possible while still maintaining the desired accuracy. Accordingly, a *lower* average prediction size is *better* (indicated by ↓).



(a) Image classification (b) Relation classification (c) Chem. prop. prediction

Figure 5-5: Few-shot CP results as a function of ϵ. The size of the prediction set of our meta CP approach is better (i.e., *smaller*) than that of our full CP baseline. Furthermore, our meta CP approach's average accuracy is close to the diagonal—allowing it to remain valid in the sense of Eq. (5.2), but also less conservative when making predictions. Note that we care more about the right-hand-side behavior of the above graphs (i.e., larger $1 - \epsilon$), as they correspond to higher coverage.

5.4.3 Baselines

For all experiments, we compare our methods to full conformal prediction, in which we use a meta-learned nonconformity scores—as defined in Eq. (5.3). Though still a straightforward application of standard conformal calibration, meta-learning $\mathcal{N}$ with auxiliary tasks already adds significant statistical power to the model over an approach that would attempt to learn $\mathcal{S}$ from scratch for each new task.

In addition to evaluating improvement over full CP, we compare our approach to other heuristics for making set valued predictions: `Top-k` and `Naive`. In `Top-k` we always take the k-highest ranked predictions. In `Naive` we select likely labels until the cumulative softmax probability exceeds $1 - \epsilon$. While related to our CP approach, we emphasize that these are only heuristics, and do not give the same theoretical performance guarantees (e.g., raw softmax probabilities are often miscalibrated).

Task	Target Acc.	Baseline CP		Meta CP		(δ, ϵ)-valid Meta CP	
	$(1 - \epsilon)$	Acc.	$\lvert\mathcal{C}_\epsilon\rvert$	Acc.	$\lvert\mathcal{M}_\epsilon\rvert$	Acc.	$\lvert\mathcal{M}_{k,\epsilon'}\rvert$
CV	0.95	1.00	10.00	0.95	3.80	0.96	3.98
	0.90	0.94	4.22	0.90	2.85	0.91	2.96
	0.80	0.83	2.38	0.80	1.89	0.81	1.95
	0.70	0.76	1.94	0.70	1.37	0.71	1.42
NLP	0.95	1.00	10.00	0.95	1.65	0.96	1.71
	0.90	0.94	1.84	0.90	1.39	0.91	1.42
	0.80	0.83	1.25	0.80	1.12	0.81	1.14
	0.70	0.76	1.10	0.70	0.93	0.71	0.94
Chem	0.95	1.00	inf	0.97	3.44	0.99	5.25
	0.90	0.94	3.28	0.92	2.62	0.95	3.02
	0.80	0.82	2.08	0.82	1.95	0.86	2.16
	0.70	0.71	1.59	0.72	1.56	0.76	1.70

Table 5.2: Few-shot CP results for specific ϵ values. We report the empirical accuracy and raw prediction set size for our two meta CP methods, and compare to our baseline CP model (full CP with meta-learned $\mathcal{N}$). For our sample-conditional meta CP approach, we fix $\delta = 0.1$ (see §5.3.4). Note that CP can produce empty sets if no labels are deemed conformal, hence the average classification size may fall below 1 for high error tolerance values of ϵ (i.e., for low $1 - \epsilon$ coverage targets).

5.5 Experimental results

In the following, we present our main conformal few-shot results. We evaluate both our sample-conditional and unconditional meta conformal prediction approaches.

Predictive efficiency. We start by testing how our meta CP approach affects the size of the prediction set. Smaller prediction set sizes correspond to more efficient conformal models. We plot prediction set size as a function of $\epsilon \in (0, 1)$ in Figure 5-5. Table 5.2 shows results for specific values of ϵ, and also shows results for our *sample-conditional* meta CP approach, where we fix $1 - \delta$ at 0.9 for all trials (note that the other meta results in Figure 5-5 and Table 5.2 are *unconditional*). Across all tasks and values of ϵ, our meta CP performs the best in terms of efficiency. Moreover, the average size of the meta CP predictions increases smoothly as a function of ϵ, while full CP suffers from discrete jumps in performance. Finally, we see that our sample-conditional $(\delta = 0.1, \epsilon)$ approach is only slightly more conservative than our

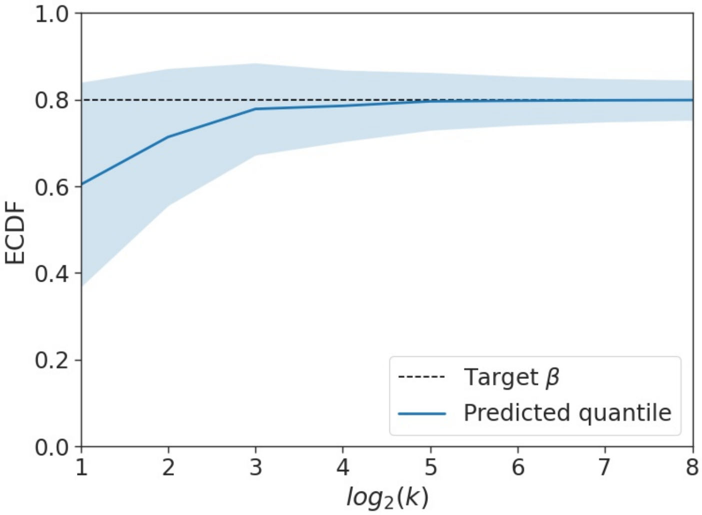


Figure 5-6: Performance of our quantile predictor $\mathcal{P}_\beta$ (for $\beta = 0.8$) on the CV task as a function of k. We use the empirical CDF over held-out points to measure the ECDF($\widehat{\mathcal{P}_\beta}$). As k increases, the predictor converges to the true quantile.

unconditional meta CP method. This is especially true for domains with a higher number of auxiliary tasks and examples per auxiliary task (i.e., CV and NLP).

Task validity. In support of Theorem 5.3.3, we observe that our meta CP approach is valid, as the test task accuracy always matches or exceeds the target performance level (this is measured on average over test task instances). Typically, meta CP is close to the target $1 - \epsilon$ level for all ϵ, which indicates that it is not overly conservative at any point (which improves the predictive efficiency). On the other hand, our full CP baseline is only close to the target accuracy when $1-\epsilon$ is near a multiple of $\frac{1}{k+1}$. This is visible from its “staircase”-like accuracy plot in Figure 5-5. We see that our sample-conditional approach is slightly conservative, as its accuracy typically exceeds $1 - \epsilon$. This is more pronounced for domains with smaller amounts of auxiliary data—which causes the adjusted $\epsilon' \leq \epsilon$ to be smaller.

Conditional coverage. Figure 5-6 shows the accuracy of our meta quantile predictor $\mathcal{P}_\beta$ as a function of k. We compute this by measuring the empirical CDF of $\mathcal{P}_\beta$'s prediction on a multiple test tasks (using a large sample of held-out data), and compare it to the target value β, i.e., ECDF($\mathcal{P}_\beta$) $\approx \beta$ if $\mathcal{P}_\beta$ is good. As expected, as k

Top-k:	CV	NLP	**Naive:**	CV		NLP	
Size (k)	Acc.	Acc.	Target Acc.	Acc.	Size	Acc.	Size
5	0.96	0.99	0.95	0.97	4.38	0.99	2.98
3	0.88	0.98	0.90	0.94	3.50	0.99	2.45
1	0.60	0.79	0.80	0.88	2.61	0.97	1.94

Table 5.3: Non-conformal baselines (for classification tasks only). `Top-k` takes a target size (k), and yields statically sized outputs. `Naive` takes a target accuracy of $1 - \epsilon$, and yields dynamically sized outputs according to softmax probability mass.

grows, $\mathcal{P}_{1-\epsilon}$ becomes more accurate. This lessens the need for large correction factors $\widehat{\Lambda}(1-\epsilon, \mathcal{I}_{\text{cal}})$, and reduces performance variance across tasks. Accordingly, as argued in Proposition 5.3.5, this also leads to task-conditional coverage asymptotically.

Baseline comparisons. Table 5.3 gives the results for our non-conformal heuristics, `Top-k` and `Naive`. We see that both approaches under-perform our CP method in terms of efficiency. Comparing to Table 5.2, we see that we achieve similar accuracy to `Top-k` with smaller average sets (while also being able to set ϵ). Similarly, `Naive` is uncalibrated and gives conservative results: for a target ϵ we obtain tighter prediction sets with our meta CP approach.

5.6 Conclusion

The ability to provide performance guarantees and make confidence-aware predictions is a critical element for many machine learning applications in the real world. Conformal prediction can afford remarkable finite-sample theoretical guarantees, but will suffer in practice when data is limited. In this chapter, we introduced a theoretically grounded approach to meta-learning few-shot conformal predictor using exchangeable collections of auxiliary tasks. Our results show that our method consistently improves performance across multiple diverse domains, and allow us to obtain meaningful and confident conformal predictors when using only a few in-task examples.

6

Generalized Conformal Risk Control

In this chapter, we extend conformal prediction to control the expected value of any monotone loss function. The algorithm generalizes split conformal prediction together with its coverage guarantee. Like conformal prediction, the conformal risk control procedure is tight up to an $\mathcal{O}(1/n)$ factor. We also introduce extensions of the idea to distribution shift, quantile risk control, multiple and adversarial risk control, and expectations of U-statistics. Worked examples from computer vision and natural language processing demonstrate the usage of our algorithm to bound the false negative rate, graph distance, and token-level F1-score.

6.1 Introduction

We seek to endow some pre-trained machine learning model with guarantees on its performance as to ensure its safe deployment. Suppose we have a base model f that is a function mapping inputs $x \in \mathcal{X}$ to values in some other space, such as a probability distribution over classes. Our job is to design a procedure that takes the output of f and post-processes it into quantities with desirable statistical guarantees.

Split conformal prediction (Vovk et al., 2005; Papadopoulos et al., 2002), which we have been working with so far (and will henceforth refer to simply as "conformal prediction"), has been useful in areas such as computer vision (Angelopoulos et al., 2021b) and natural language processing (Fisch et al., 2021e) to provide such a guarantee. By measuring the model's performance on a *calibration dataset* $\left\{(X_i, Y_i)\right\}_{i=1}^{n}$ of feature-response pairs, conformal prediction post-processes the model to construct prediction sets that bound the *miscoverage*,

$$\mathbb{P}\Big(Y_{n+1} \notin \mathcal{C}(X_{n+1})\Big) \leq \alpha, \tag{6.1}$$

where (X_{n+1}, Y_{n+1}) is a new test point, α is a user-specified error rate (e.g., 10%), and $\mathcal{C}$ is a function of the model and calibration data that outputs a prediction set.[1] Note that $\mathcal{C}$ is formed using the first n data points, and the probability in Eq. (6.1) is over the randomness in all $n+1$ data points (i.e., the draw of both the calibration and test points). In this chapter, we extend conformal prediction to prediction tasks where the natural notion of error is not simply miscoverage. In particular, our main result is that a generalization of conformal prediction provides guarantees of the form

$$\mathbb{E}\Big[\ell\Big(\mathcal{C}(X_{n+1}), Y_{n+1}\Big)\Big] \leq \alpha, \tag{6.2}$$

for any bounded *loss function* ℓ that shrinks as $\mathcal{C}(X_{n+1})$ grows. We call this a *conformal risk control* guarantee. Note that Eq. (6.2) recovers the conformal miscoverage guarantee in Eq. (6.1) when using the miscoverage loss,

$$\ell\Big(\mathcal{C}(X_{n+1}), Y_{n+1}) = \mathbf{1}\{Y_{n+1} \notin \mathcal{C}(X_{n+1})\} \tag{6.3}$$

However, our algorithm also extends conformal prediction to situations where other loss functions, such as the false negative rate (FNR), are more appropriate.

As an example, consider multilabel classification, where the $Y_i \subseteq \{1, ..., K\}$ are sets comprising a subset of K classes. Given a trained multilabel classifier $f : \mathcal{X} \to [0, 1]^K$, we want to output sets that include a large fraction of the true classes in Y_i. To that end, we post-process the model's raw outputs into the set of classes with sufficiently

[1] For conformal prediction, α plays the same role as ϵ per notation in previous sections.

high scores, $\mathcal{C}_\lambda(x) = \{k : f(X)_k \geq 1 - \lambda\}$. Note that as the threshold λ grows, we include more classes in $\mathcal{C}_\lambda(x)$—i.e., it becomes more conservative. In this case, conformal risk control finds a threshold value $\hat{\lambda}$ that controls the fraction of missed classes, i.e., the expected value of $\ell\Big(\mathcal{C}_{\hat{\lambda}}(X_{n+1}), Y_{n+1}\Big) = 1 - |Y_{n+1} \cap \mathcal{C}_\lambda(X_{n+1})|/|Y_{n+1}|$. Setting $\alpha = 0.1$ would ensure that our algorithm produces sets $\mathcal{C}_{\hat{\lambda}}(X_{n+1})$ containing $\geq 90\%$ of the true classes in Y_{n+1} on average.

6.1.1 Algorithm and preview of main results

Formally, we will consider post-processing the predictions of the model f to create a function $\mathcal{C}_\lambda(\cdot)$. The function has a parameter λ that encodes its level of conservativeness: larger λ values yield more conservative outputs (e.g., larger prediction sets). To measure the quality of the output of $\mathcal{C}_\lambda$, we consider a loss function $\ell(\mathcal{C}_\lambda(x), y) \in (-\infty, B]$ for some $B < \infty$. We require the loss function to be non-increasing as a function of λ. Our goal is to choose $\hat{\lambda}$ based on the observed data $\Big\{(X_i, Y_i)\Big\}_{i=1}^{n}$ so that risk control as in Eq. (6.2) holds.

We now rewrite this same task in a more notationally convenient and abstract form. Consider an exchangeable collection of non-increasing, random functions $L_i : \Lambda \to (-\infty, B]$, $i = 1, \ldots, n+1$. Throughout the paper, we assume $\lambda_{\max} \triangleq \sup \Lambda \in \Lambda$. We seek to use the first n functions to choose a value of the parameter, $\hat{\lambda}$, in such a way that the risk on the unseen function is controlled:

$$\mathbb{E}\Big[L_{n+1}\Big(\hat{\lambda}\Big)\Big] \leq \alpha. \tag{6.4}$$

We are primarily motivated by the case where $L_i(\lambda) = \ell(\mathcal{C}_\lambda(X_i), Y_i)$, in which case the guarantee in Eq. (6.4) coincides with risk control as in Eq. (6.2).

Now we describe the algorithm. Let $\widehat{R}_n(\lambda) = (L_1(\lambda) + \ldots + L_n(\lambda))/n$. Given any desired risk level upper bound $\alpha \in (-\infty, B)$, define

$$\hat{\lambda} = \inf\left\{\lambda : \frac{n}{n+1}\widehat{R}_n(\lambda) + \frac{B}{n+1} \leq \alpha\right\}. \tag{6.5}$$

When the set is empty, we define $\hat{\lambda} = \lambda_{\max}$. Our proposed *conformal risk control* algorithm is to deploy $\hat{\lambda}$ on the forthcoming test point. Our main result is that this algorithm satisfies Eq. (6.4). When the L_i are i.i.d. from a continuous distribution, the algorithm satisfies a tight lower bound saying it is not too conservative,

$$\mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}\big)\Big] \geq \alpha - \frac{2B}{n+1}. \tag{6.6}$$

We show the reduction from conformal risk control to conformal prediction in §6.3.3. Furthermore, if the risk is non-monotone, then this algorithm does not control the risk; we discuss this in §6.3.4. Finally, we provide both practical examples and several theoretical extensions of our procedure in §6.4 and §6.5, respectively.

6.2 Related work

As previously discussed, here we primarily build on *split conformal prediction* (Papadopoulos et al., 2002); statistical properties of this algorithm including the coverage upper bound were studied in (Lei et al., 2018). Recently there have been many extensions of the conformal algorithm, mainly targeting deviations from exchangeability (Tibshirani et al., 2019; Gibbs and Candes, 2021; Barber et al., 2022; Fannjiang et al., 2022) and improved conditional coverage (Barber et al., 2020; Romano et al., 2019a; Guan, 2020; Romano et al., 2020b; Angelopoulos et al., 2021b). Most relevant to us is recent work on risk control in high probability (Vovk, 2012; Bates et al., 2020; Angelopoulos et al., 2021a) and its applications (Park et al., 2020; Schuster et al., 2021b, 2022; Sankaranarayanan et al., 2022; Angelopoulos et al., 2022b,c). Though these works closely relate to ours in terms of motivation, the algorithm presented herein differs greatly: it has a guarantee in expectation, and neither the algorithm nor its analysis share much technical similarity with these previous works.

To elaborate on the difference between our work and previous literature, first consider conformal prediction. The purpose of conformal prediction is to provide coverage guarantees of the form in Eq. (6.1). The guarantee available through conformal risk control, Eq. (6.4), strictly subsumes that of conformal prediction; it is generally impossible to recast risk control as coverage control. As a second question, one might ask whether Eq. (6.4) can be achieved through standard statistical machinery, such as uniform concentration inequalities. Though it is possible to integrate a uniform concentration inequality to get a bound in expectation, this strategy tends to be excessively loose both in theory and in practice (see, e.g., the bound of Anthony and Shawe-Taylor (1993)). The technique herein avoids these complications; it is simpler than concentration-based approaches, practical to implement, and tight up to a factor of $1/n$, which is comparatively faster than concentration would allow. Finally, we target distribution-free finite-sample control of Eq. (6.4), but as a side-note it is also worth pointing the reader to the rich literature on functional central limit theorems (Davidson and De Jong, 2000), which are another way of estimating risk functions.

6.3 Conformal risk control

In this section, we establish the core theoretical properties of conformal risk control. All proofs, unless otherwise specified, are deferred to Appendix F.1.

6.3.1 Risk control

We first show that the proposed algorithm leads to risk control when the loss is monotone.

Theorem 6.3.1. *Assume that $L_i(\lambda)$ is non-increasing in λ, right-continuous, and*

$$L_i(\lambda_{\max}) \leq \alpha, \quad \sup_{\lambda} L_i(\lambda) \leq B < \infty \text{ almost surely.} \tag{6.7}$$

Then

$$\mathbb{E}[L_{n+1}(\hat{\lambda})] \leq \alpha. \tag{6.8}$$

Proof. Let $\widehat{R}_{n+1}(\lambda) = (L_1(\lambda) + \ldots + L_{n+1}(\lambda))/(n+1)$ and

$$\hat{\lambda}' = \inf\left\{\lambda \in \Lambda : \widehat{R}_{n+1}(\lambda) \leq \alpha\right\}. \tag{6.9}$$

Since $\inf_\lambda L_i(\lambda) = L_i(\lambda_{\max}) \leq \alpha$, $\hat{\lambda}'$ is well-defined almost surely. Since $L_{n+1}(\lambda) \leq B$, we know $\widehat{R}_{n+1}(\lambda) = \frac{n}{n+1}\widehat{R}_n(\lambda) + \frac{L_{n+1}(\lambda)}{n+1} \leq \frac{n}{n+1}\widehat{R}_n(\lambda) + \frac{B}{n+1}$. Thus,

$$\frac{n}{n+1}\widehat{R}_n(\lambda) + \frac{B}{n+1} \leq \alpha \Longrightarrow \widehat{R}_{n+1}(\lambda) \leq \alpha. \tag{6.10}$$

This implies $\hat{\lambda}' \leq \hat{\lambda}$ when the LHS holds for some $\lambda \in \Lambda$. When the LHS is above α for all $\lambda \in \Lambda$, by definition, $\hat{\lambda} = \lambda_{\max} \geq \hat{\lambda}'$. Thus, $\hat{\lambda}' \leq \hat{\lambda}$ almost surely. Since $L_i(\lambda)$ is non-increasing in λ,

$$\mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}\big)\Big] \leq \mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}'\big)\Big]. \tag{6.11}$$

Let E be the multiset of loss functions $\{L_1, \ldots, L_{n+1}\}$. Then $\hat{\lambda}'$ is a function of E, or, equivalently, $\hat{\lambda}'$ is a constant conditional on E. Additionally, $L_{n+1}(\lambda)|E \sim \text{Uniform}(\{L_1, ..., L_{n+1}\})$ by exchangeability. These facts combined with the right-continuity of L_i imply

$$\mathbb{E}\left[L_{n+1}(\hat{\lambda}') \mid E\right] = \frac{1}{n+1}\sum_{i=1}^{n+1} L_i(\hat{\lambda}') \leq \alpha. \tag{6.12}$$

The proof is completed by the law of total expectation and Eq. (6.11). □

6.3.2 A tight risk lower bound

Next we show that the conformal risk control procedure is tight up to a factor $2B/(n+1)$ that cannot be improved in general. Like the standard conformal coverage upper bound, the proof will rely on a form of continuity that prohibits large jumps in the risk function. Towards that end, we will define the *jump function* below, which quantifies the size of the discontinuity in a right-continuous input function l at point λ:

$$J(l, \lambda) = \lim_{\epsilon \to 0^+} l(\lambda - \epsilon) - l(\lambda) \tag{6.13}$$

The jump function measures the size of a discontinuity at $l(\lambda)$. When there is a discontinuity and l is non-increasing, $J(l, \lambda) > 0$. When there is no discontinuity, the jump function is zero. The next theorem will assume that the probability that L_i has a discontinuity at any pre-specified λ is $\mathbb{P}(J(L_i, \lambda) > 0) = 0$. Under this assumption the conformal risk control procedure is not too conservative.

Theorem 6.3.2. *In the setting of Theorem 6.3.1, further assume that the L_i are i.i.d., $L_i \geq 0$, and for any λ, $\mathbb{P}\left(J(L_i, \lambda) > 0\right) = 0$. Then*

$$\mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}\big)\Big] \geq \alpha - \frac{2B}{n+1}. \tag{6.14}$$

Proof. See Appendix F.1.1. □

This bound is tight for general monotone loss functions, as we show next.

Proposition 6.3.3. *In the setting of Theorem 6.3.2, for any $\epsilon > 0$, there exists a loss function and $\alpha \in (0, 1)$ such that*

$$\mathbb{E}\left[L_{n+1}\big(\hat{\lambda}\big)\right] \leq \alpha - \frac{2B - \epsilon}{n+1}. \tag{6.15}$$

Proof. See Appendix F.1.2. □

Since we can take ϵ arbitrarily close to zero, we conclude that the factor $2B/(n+1)$ in Theorem 6.3.2 is required in the general case.

6.3.3 Conformal prediction reduces to risk control

Conformal prediction can be thought of as controlling the expectation of an indicator loss function. Recall that the risk upper bound Eq. (6.2) specializes to the conformal coverage guarantee in Eq. (6.1) when the loss function is the indicator of a miscoverage event. The conformal risk control procedure specializes to conformal prediction under this loss function as well. However, the risk lower bound in Theorem 6.3.2 has a slightly worse constant than the usual conformal guarantee. We now describe these correspondences.

First, we show the equivalence of the algorithms. In conformal prediction, we have conformal scores $s(X_i, Y_i)$ for some score function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Based on this score function, we create prediction sets for the test point X_{n+1} as

$$\mathcal{C}_{\hat{\lambda}}(X_{n+1}) = \Big\{ y : s(X_{n+1}, y) \leq \hat{\lambda} \Big\}, \tag{6.16}$$

where $\hat{\lambda}$ is the conformal quantile, a parameter that is set based on the calibration data. In particular, conformal prediction chooses $\hat{\lambda}$ to be the $\lceil (n+1)(1-\alpha) \rceil / n$ sample quantile of $\{s(X_i, Y_i)\}_{i=1}^n$. To formulate this in the language of risk control, we consider a *miscoverage loss* $L_i^{\text{Cvg}}(\lambda) = \mathbf{1}\{Y_i \notin \widehat{\mathcal{C}}_\lambda(X_i)\} = \mathbf{1}\{s(X_i, Y_i) > \lambda\}$. Direct calculation of $\hat{\lambda}$ from Eq. (6.5) then shows the equivalence of the proposed procedure to conformal prediction:

$$\begin{aligned} \hat{\lambda} &= \inf \left\{ \lambda \colon \frac{1}{n+1} \sum_{i=1}^n \mathbf{1}\{s(X_i, Y_i) > \lambda\} + \frac{1}{n+1} \leq \alpha \right\} \\ &= \underbrace{\inf \left\{ \lambda : \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{s(X_i, Y_i) \leq \lambda\} \geq \frac{\lceil (n+1)(1-\alpha) \rceil}{n} \right\}}_{\text{conformal prediction algorithm}}. \end{aligned} \tag{6.17}$$

Next, we discuss how the risk lower bound relates to its conformal prediction equivalent. In the setting of conformal prediction, Lei et al. (2018) proves that $\mathbb{P}(Y_{n+1} \notin \mathcal{C}_{\hat{\lambda}}(X_{n+1})) \geq \alpha - 1/(n+1)$ when the conformal score function follows a continuous distribution. Theorem 6.3.2 recovers this guarantee with a slightly worse constant: $\mathbb{P}(Y_{n+1} \notin \mathcal{C}_{\hat{\lambda}}(X_{n+1})) \geq \alpha - 2/(n+1)$. First, note that our assumption in Theorem 6.3.2 about the distribution of discontinuities specializes to the continuity of the score function when the miscoverage loss is used:

$$\mathbb{P}\left(J\Big(L_i^{\text{Cvg}}, \lambda\Big) > 0 \right) = 0 \iff \mathbb{P}(s(X_i, Y_i) = \lambda) = 0. \tag{6.18}$$

However, the bound for the conformal case is better than the bound for the general case in Theorem 6.3.2 by a factor of two, which cannot be improved according to Proposition 6.3.3. The fact that conformal prediction has a slightly tighter lower bound than conformal risk control is an interesting oddity of the binary loss function; however, it is of little practical importance, as the difference between $1/(n+1)$ and $2/(n+1)$ is small even for moderate values of n.

6.3.4 Controlling general loss functions

We next show that the conformal risk control algorithm does *not* control the risk if the L_i are not assumed to be monotone. In particular, Eq. (6.4) does not hold. We show this by example.

Proposition 6.3.4. *For any ϵ, there exists a non-monotone loss function such that*

$$\mathbb{E}\left[L_{n+1}\left(\hat{\lambda}\right)\right] \geq B - \epsilon. \tag{6.19}$$

Proof. See Appendix F.1.3. □

Notice that for any desired level α, the expectation in Eq. (6.4) can be arbitrarily close to B. Since the function values here are in $[0, B]$, this means that even for bounded random variables, risk control can be violated by an arbitrary amount—unless further assumptions are placed on the L_i. However, the algorithms developed may still be appropriate for near-monotone loss functions. Simply 'monotonizing' all loss functions L_i and running conformal risk control will guarantee Eq. (6.4), but this strategy will only be powerful if the loss is near-monotone. For concreteness, we describe this procedure below as a corollary of Theorem 6.3.1.

Corollary 6.3.5. *Allow $L_i(\lambda)$ to be any (possibly non-monotone) function of λ satisfying Eq. (6.7). Take $\tilde{L}_i(\lambda) = \sup\limits_{\lambda' \geq \lambda} L_i(\lambda')$, $\tilde{R}_n(\lambda) = \frac{1}{n}\sum\limits_{i=1}^{n} \tilde{L}_i(\lambda)$, and*

$$\tilde{\lambda} = \inf\left\{\lambda : \frac{n}{n+1}\tilde{R}_n(\lambda) + \frac{B}{n+1} \leq \alpha\right\}. \tag{6.20}$$

Then,

$$\mathbb{E}\left[L_{n+1}\left(\tilde{\lambda}\right)\right] \leq \alpha. \tag{6.21}$$

Proof. This is a direct application of Theorem 6.3.1 to the new loss $\tilde{L}_i$. □

If the loss function is already monotone, then $\tilde{\lambda}$ reduces to $\hat{\lambda}$.

We next show that the proposed algorithm leads to asymptotic risk control for non-monotone risk functions when applied to a monotone version of the empirical risk (which is less strict than the monotone version of each individual loss function as handled previously). However, this algorithm again is only powerful when the risk is near-monotone and reduces to the standard conformal risk control algorithm when the loss is monotone. We set the *monotonized empirical risk* to be

$$\widehat{R}_n^{\uparrow}(\lambda) = \sup_{t \geq \lambda} \widehat{R}_n(t), \tag{6.22}$$

then define

$$\hat{\lambda}_n^{\uparrow} = \inf\left\{\lambda : \widehat{R}_n^{\uparrow}(\lambda) \leq \alpha\right\}. \tag{6.23}$$

Theorem 6.3.6. *Let the $L_i(\lambda)$ be right-continuous, i.i.d., bounded (both above and below) functions satisfying* (6.7)*. Then,*

$$\lim_{n\to\infty} \mathbb{E}\left[L_{n+1}\left(\hat{\lambda}_n^{\uparrow}\right)\right] \leq \alpha. \tag{6.24}$$

Proof. See Appendix F.1.4. □

Theorem 6.3.6 implies that an analogous procedure to 6.5 also controls the risk asymptotically. In particular, taking

$$\tilde{\lambda}^{\uparrow} = \inf\left\{\lambda : \widehat{R}_n^{\uparrow}(\lambda) + \frac{B}{n+1} \leq \alpha\right\} \tag{6.25}$$

also results in asymptotic risk control (to see this, plug $\tilde{\lambda}^{\uparrow}$ into Theorem 6.3.6 and see that the risk level is bounded above by $\alpha - \frac{B}{n+1}$). Note that in the case of a monotone loss function, $\tilde{\lambda}^{\uparrow} = \hat{\lambda}$. However, the counterexample in Proposition 6.3.4 does not apply to $\tilde{\lambda}^{\uparrow}$, and it is currently unknown whether this procedure does or does not provide finite-sample risk control.

6.4 Examples

To demonstrate the flexibility and empirical effectiveness of the proposed algorithm, we apply it to four tasks across computer vision and natural language processing. All four loss functions are non-binary, monotone losses bounded by 1. They are commonly used within their respective application domains. Our results validate that the procedure bounds the risk as desired and gives useful outputs to the end-user. We note that the choices of $\mathcal{C}_\lambda$ used herein are *only for the purposes of illustration*; any nested family of sets will work. For each example use case, for a representative α (details provided for each task) we provide both qualitative results, as well as quantitative histograms of the risk and set sizes over 1000 random data splits that demonstrate valid risk control (i.e., with mean $\leq \alpha$).

6.4.1 FNR control in tumor segmentation

In the tumor segmentation setting, our input is a $d \times d$ image and our label is a set of pixels $Y_i \in \wp\left(\{(1,1),(1,2),...,(d,d)\}\right)$, where $\wp$ denotes the power set. We build on an image segmentation model $f : \mathcal{X} \to [0,1]^{d\times d}$ outputting a probability for each pixel and measure loss as the fraction of false negatives,

$$L_i^{\mathrm{FNR}}(\lambda) = 1 - \frac{|Y_i \cap \mathcal{C}_\lambda(X_i)|}{|Y_i|}, \text{ where } \mathcal{C}_\lambda(X_i) = \{y : f(X_i)_y \geq 1 - \lambda\}. \qquad (6.26)$$

The expected value of L_i^{FNR} is the FNR. Since L_i^{FNR} is monotone, so is the FNR. Thus, we use the technique in Section 6.3.1 to pick $\hat{\lambda}$ by Eq. (6.5) that controls the

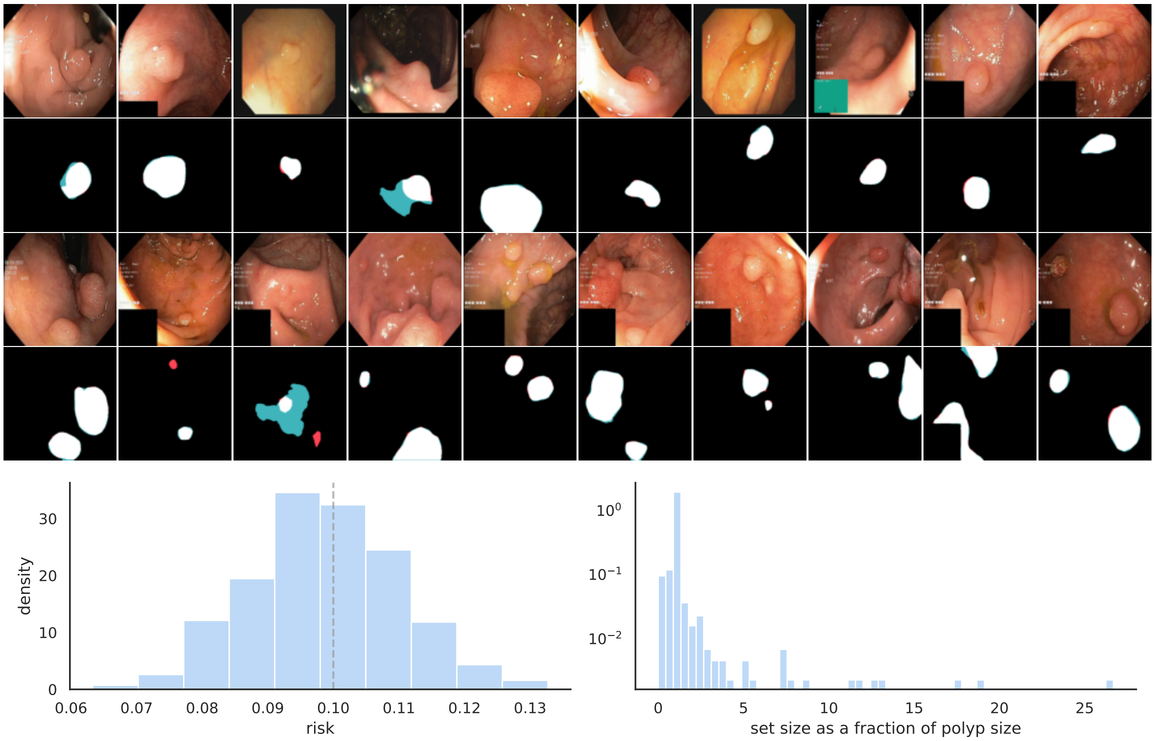


Figure 6-1: **FNR control in tumor segmentation**. The top figure shows examples of our procedure with correct pixels in white, false positives in blue, and false negatives in red. The bottom plots report FNR and set size over 1000 independent random data splits. The dashed gray line marks α.

FNR on a new point, resulting in the following guarantee:

$$\mathbb{E}\Big[L_{n+1}^{\text{FNR}}(\hat{\lambda})\Big] \leq \alpha. \tag{6.27}$$

For evaluating the proposed procedure we pool data from several online open-source gut polyp segmentation datasets: Kvasir, Hyper-Kvasir, CVC-ColonDB, CVC-ClinicDB, and ETIS-Larib. We choose a PraNet (Fan et al., 2020b) as our base model f and used $n = 1000$, and evaluated risk control with the 781 remaining validation data points. We report results with $\alpha = 0.1$ in Figure 6-1. The mean and standard deviation of the risk over 1000 trials are 0.0987 and 0.0114, respectively.

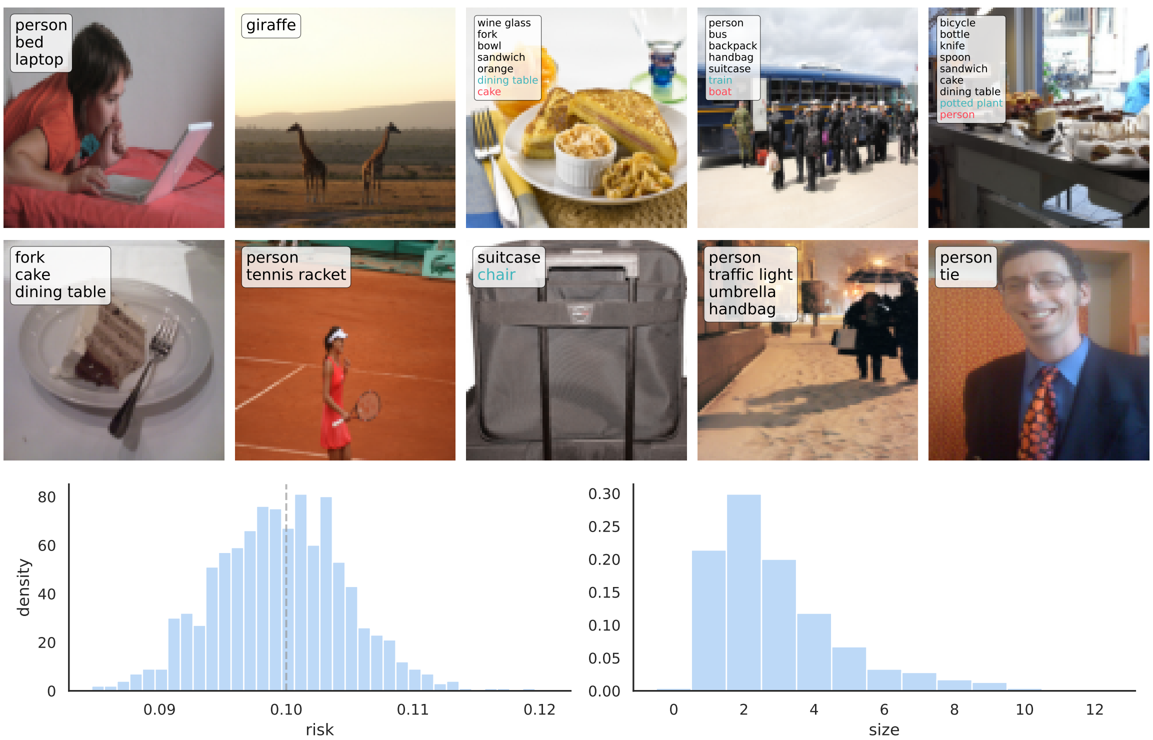


Figure 6-2: **FNR control on MS COCO**. The top figure shows examples with correct classes in black, false positives in blue, and false negatives in red. The bottom plots report our false negative rate loss and set size for multilabel classification over 1000 independent random data splits. The dashed gray line marks α.

6.4.2 FNR control in multilabel classification

In the multilabel classification setting, our input X_i is an image and our label is a set of classes $Y_i \subset \{1, \dots, K\}$ for some number of classes K. Using a multiclass classification model $f : \mathcal{X} \to [0, 1]^K$, we form prediction sets and calculate the number of false positives exactly as in Eq. (6.26). By Theorem 6.3.1, picking $\hat{\lambda}$ as in Eq. (6.5) again yields the FNR-control guarantee in Eq. (6.27).

We use the Microsoft Common Objects in Context (MS COCO) computer vision dataset (Lin et al., 2014), a large-scale 80-class multiclass classification baseline dataset commonly used in computer vision, to evaluate the proposed procedure. We choose a TResNet (Ridnik et al., 2020) as our base model f and used $n = 4000$, and evaluated risk control with 1000 validation data points. We report results with $\alpha = 0.1$ in Figure 6-2. The mean and standard deviation of the risk over 1000 trials

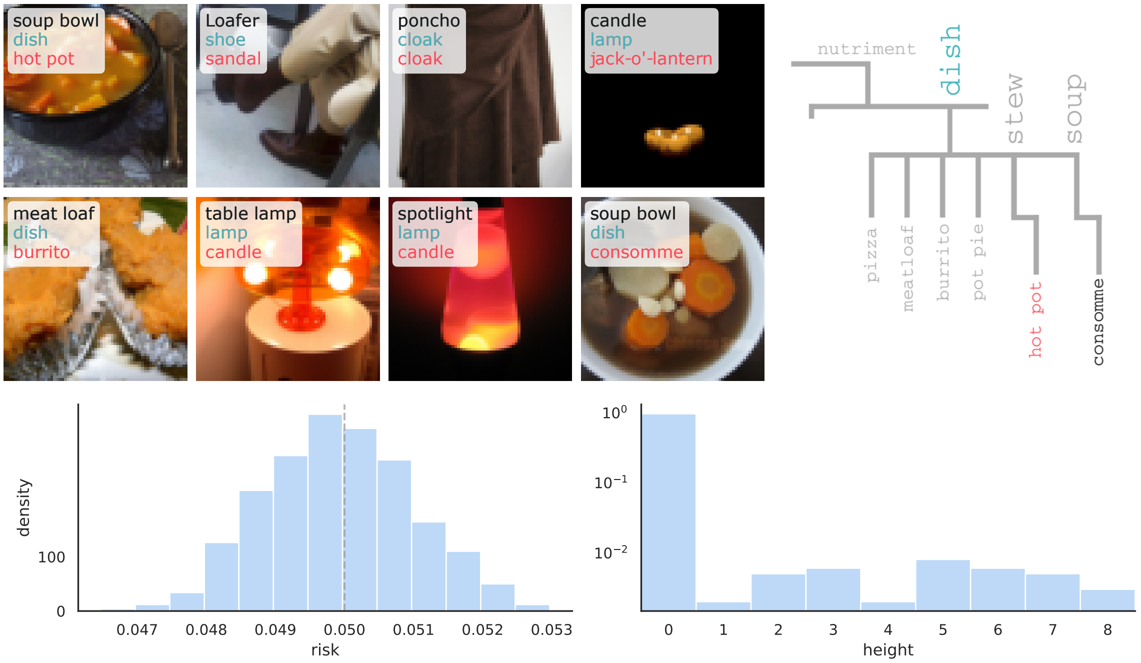


Figure 6-3: **Control of graph distance on hierarchical ImageNet**. The top figure shows examples with correct classes in black, false positives in blue, and false negatives in red. The bottom plots report our minimum hierarchical distance loss and set size over 1000 independent random data splits. The dashed gray line marks α.

are 0.0996 and 0.0052, respectively.

6.4.3 Control of graph distance in hierarchical image classification

In the K-class hierarchical classification setting, our input X_i is an image and our label is a leaf node $Y_i \in \{1, ..., K\}$ on a tree with nodes $\mathcal{V}$ and edges $\mathcal{E}$. Using a single-class classification model $f : \mathcal{X} \to \Delta^K$, we calibrate a loss in graph distance between the interior node we select and the closest ancestor of the true class. For any $x \in \mathcal{X}$, let $\hat{y}(x) = \arg\max_k f(x)_k$ be the class with the highest estimated probability. Further, let $d : \mathcal{V} \times \mathcal{V} \to \mathbb{Z}$ be the function that returns the length of the shortest path between two nodes, let $\mathcal{A} : \mathcal{V} \to 2^{\mathcal{V}}$ be the function that returns the ancestors of its argument, and let $\mathcal{P} : \mathcal{V} \to 2^{\mathcal{V}}$ be the function that returns the set of leaf nodes that are descendants of its argument. We also let

$$g(v, x) = \sum_{k \in \mathcal{P}(v)} f(x)_k \tag{6.28}$$

be the sum of scores of leaves descended from v. Define a hierarchical distance

$$d_H(v,u) = \inf_{a \in \mathcal{A}(v)} \{d(a,u)\}. \tag{6.29}$$

For a set of nodes $\mathcal{C}_\lambda \in 2^{\mathcal{V}}$, we then define the set-valued loss

$$L_i^{\text{Graph}}(\lambda) = \inf_{s \in \mathcal{C}_\lambda(X_i)} \{d_H(y,s)\}/D, \text{ where } \mathcal{C}_\lambda(x) = \bigcap_{\{a \in \mathcal{A}(\hat{y}(x))\ :\ g(a,x) \geq -\lambda\}} \mathcal{P}(a). \tag{6.30}$$

This loss returns zero if y is a child of any element in $\mathcal{C}_\lambda$, and otherwise returns the minimum distance between any element of $\mathcal{C}_\lambda$ and any ancestor of y, scaled by the depth D. Thus, it is a monotone loss function and can be controlled by choosing $\hat{\lambda}$ as in Eq. (6.5) to achieve the guarantee

$$\mathbb{E}\left[L_{n+1}^{\text{Graph}}(\hat{\lambda})\right] \leq \alpha. \tag{6.31}$$

We use the ImageNet dataset (Deng et al., 2009), which comes with an existing label hierarchy, WordNet, of maximum depth $D = 14$. We choose a ResNet152 (He et al., 2016) as our base model f and used $n = 30k$, and evaluated risk control with the remaining $20k$. We report results with $\alpha = 0.05$ in Figure 6-3. The mean and standard deviation of the risk over 1000 trials are 0.0499 and 0.0011, respectively.

6.4.4 F1-score control in open-domain question answering

In the open-domain question answering setting, our input X_i is a question and our label Y_i is a set of (possibly non-unique) correct answers. For example, the input

$$X_{n+1} = \text{“Where was Barack Obama Born?”}$$

could have the answer set

$$Y_{n+1} = \{\text{“Hawaii”, “Honolulu, Hawaii”, “Kapo'olani Medical Center”}\}$$

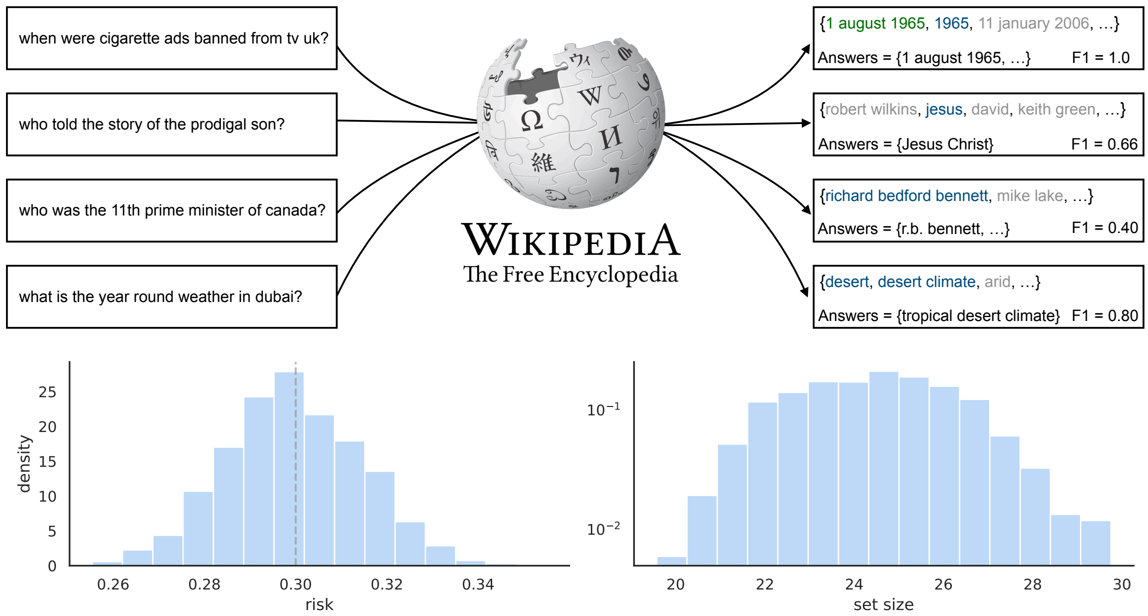


Figure 6-4: **F1-score control on Natural Questions**. The top figure shows examples with fully correct answers in green, partially correct answers in blue, and false positives in gray. Due to the nature of the evaluation, answers that are technically correct may still be down-graded if they do not match the reference (we treat this as part of the randomness in the task). The bottom plots report the F1 risk and average set size over 1000 independent random data splits. The dashed gray line marks α.

Here we treat all questions and answers as being composed of sequences (up to size m) of tokens in a vocabulary $\mathcal{V}$—i.e., assuming k valid answers, we have $X_i \in \mathcal{Z}$ and $Y_i \in \mathcal{Z}^k$, where $\mathcal{Z} := \mathcal{V}^m$. Using an open-domain question answering model that individually scores candidate output answers $f\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, we calibrate the *best* token-based F1-score of the prediction set, taken over all pairs of predictions and answers:

$$\begin{gathered} L_i^{\mathrm{F1}}(\lambda) = 1 - \max \Big\{ \mathrm{F1}(a, c) \colon c \in \mathcal{C}_\lambda(X_i), a \in Y_i \Big\}, \\ \text{where } \mathcal{C}_\lambda(X_i) = \{ y \in \mathcal{V}^m : f(X_i, y) \geq \lambda \}\,. \end{gathered} \tag{6.32}$$

We define the F1-score following popular QA evaluation metrics (Rajpurkar et al., 2016), where we treat predictions and ground truth answers as bags of tokens and compute the geometric average of their precision and recall (while ignoring punctuation and articles {“a”, “an”, “the”}). Since L_i^{F1}, as defined in this way, is monotone and upper bounded by 1, it can be controlled by choosing $\hat{\lambda}$ as in Section 6.3.1 to achieve the following guarantee:

$$\mathbb{E}\left[L_{n+1}^{\mathrm{F1}}(\hat{\lambda})\right] \leq \alpha. \tag{6.33}$$

We use the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019), a popular open-domain question answering baseline, to evaluate our method. We use the splits distributed as part of the Dense Passage Retrieval (DPR) package (Karpukhin et al., 2020). Our base model is the DPR Retriever-Reader model (Karpukhin et al., 2020), which retrieves passages from Wikipedia that might contain the answer to the given query, and then uses a reader model to extract text sub-spans from the retrieved passages that serve as candidate answers. Instead of enumerating all possible answers to a given question (which is intractable), we retrieve the top several hundred candidate answers, extracted from the top 100 passages (which is sufficient to control all risks of interest). We use $n = 2500$ calibration points, and evaluate risk control with the remaining 1110. We report results with $\alpha = 0.3$ (chosen empirically as the lowest F1 score which typically results in nearly correct answers) in Figure 6-4. The mean and standard deviation of the risk over 1000 trials are 0.2996 and 0.0150, respectively.

6.5 Extensions

In this section, we discuss several theoretical extensions of our procedure.

6.5.1 Risk control under distributional shift

Suppose the researcher wants to control the risk under a distribution shift. Then the goal in Eq. (6.4) can be redefined as

$$\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)\sim P_{\text{train}},\,(X_{n+1},Y_{n+1})\sim P_{\text{test}}}\left[L_{n+1}\left(\hat{\lambda}\right)\right] \leq \alpha, \tag{6.34}$$

where P_{test} denotes the test distribution that is different from the training distribution P_{train} that $(X_i, Y_i)_{i=1}^n$ are sampled from. Assuming that P_{test} is absolutely continuous with respect to P_{train}, the weighted objective (6.34) can be rewritten as

$$\mathbb{E}_{(X_1,Y_1),\ldots,(X_{n+1},Y_{n+1})\sim P_{\text{train}}}\left[w(X_{n+1},Y_{n+1})L_{n+1}\left(\hat{\lambda}\right)\right] \leq \alpha, \\ \text{where } w(x,y) = \frac{dP_{\text{test}}(x,y)}{dP_{\text{train}}(x,y)}. \tag{6.35}$$

When w is known and bounded, we can apply our procedure on the loss function

$$\tilde{L}_{n+1}(\lambda) = w(X_{n+1},Y_{n+1})L_{n+1}(\lambda), \tag{6.36}$$

which is non-decreasing, bounded, and right-continuous in λ whenever L_{n+1} is. Thus, Theorem 6.3.1 guarantees that the resulting $\hat{\lambda}$ satisfies Eq. (6.35).

In the setting of transductive learning, X_{n+1} is available to the user. If the conditional distribution of Y given X remains the same in the training and test domains, the distributional shift reduces to a covariate shift and

$$w(X_{n+1},Y_{n+1}) = w(X_{n+1}) \triangleq \frac{dP_{\text{test}}(X_{n+1})}{dP_{\text{train}}(X_{n+1})}. \tag{6.37}$$

In this case, we can achieve the risk control even when w is unbounded. In particular, assuming $L_i \in [0,B]$, for any potential value x of the covariate, we define

$$\hat{\lambda}(x) = \inf\left\{\lambda : \frac{\sum_{i=1}^{n} w(X_i)L_i(\lambda) + w(x)B}{\sum_{i=1}^{n} w(X_i) + w(x)} \leq \alpha\right\}. \tag{6.38}$$

When λ does not exist, we simply set $\hat{\lambda}(x) = \max\Lambda$. It is not hard to see that $\hat{\lambda}(x) \equiv \hat{\lambda}$ in the absence of covariate shifts. We can prove the following result.

Proposition 6.5.1. *In the setting of Theorem 6.3.1,*

$$\mathbb{E}_{(X_1,Y_1),\ldots,(X_n,Y_n)\sim P_{\text{train}},(X_{n+1},Y_{n+1})\sim P_{\text{test}}}[L_{n+1}(\hat{\lambda}(X_{n+1}))] \leq \alpha. \tag{6.39}$$

Proof. See Appendix F.1.5. □

It is easy to show that the weighted conformal procedure (Tibshirani et al., 2019) is a special case with $L_i(\lambda) = \mathbf{1}\{Y_i \notin \mathcal{C}_\lambda(X_i)\}$ where $\mathcal{C}_\lambda(X_i)$ is the prediction set that thresholds the conformity score at λ. Thus, Proposition 6.5.1 generalizes Tibshirani et al. (2019) to any monotone risk. When the covariate shift $w(x)$ is unknown but unlabeled data in the test domain are available, it can be estimated, up to a multiplicative factor that does not affect $\hat{\lambda}(x)$, by any probabilistic classification algorithm; see Lei and Candès (2020) and Candès et al. (2023) in the context of missing and censored data, respectively. We leave the full investigation of weighted conformal risk control with an estimated covariate shift for future research.

Total variation bound

Finally, for arbitrary distribution shifts, we give a total variation bound describing the way standard (unweighted) conformal risk control degrades. The bound is analogous to that of Barber et al. (2022) for independent but non-identically distributed data (see their Section 4.1), though the proof is different. Here we will use the notation $Z_i = (X_i, Y_i)$, and $\hat{\lambda}(Z_1, \ldots, Z_n)$ to refer to that chosen in Eq. (6.5).

Proposition 6.5.2. *Let $Z = (Z_1, \ldots, Z_{n+1})$ be a sequence of random variables. Then, under the conditions in Theorem 6.3.1,*

$$\mathbb{E}\left[L_{n+1}(\hat{\lambda})\right] \leq \alpha + B\sum_{i=1}^{n} \mathrm{TV}(Z_i, Z_{n+1}). \tag{6.40}$$

If further the assumptions of Theorem 6.3.2 hold,

$$\mathbb{E}\left[L_{n+1}(\hat{\lambda})\right] \geq \alpha - B\left(\frac{2}{n+1} + \sum_{i=1}^{n} \mathrm{TV}(Z_i, Z_{n+1})\right). \tag{6.41}$$

Proof. See Appendix F.1.6. □

6.5.2 Quantile risk control

Snell et al. (2022) generalizes Bates et al. (2020) to control the quantile of a monotone loss function conditional on $(X_i, Y_i)_{i=1}^n$ with probability $1-\delta$ over the calibration dataset for any user-specified tolerance parameter δ. In some applications, it may be sufficient to control the unconditional quantile of the loss function, which alleviates the burden of the user to choose the tolerance parameter δ.

For any random variable X, let

$$\text{Quantile}_{\beta}(X) = \inf\{x : \mathbb{P}(X \leq x) \geq \beta\}. \tag{6.42}$$

Analogous to (6.4), we want to find $\hat{\lambda}$ based on $(X_i, Y_i)_{i=1}^{n}$ such that

$$\text{Quantile}_{\beta}\left(L_{n+1}(\hat{\lambda}_{\beta})\right) \leq \alpha. \tag{6.43}$$

By definition,

$$\text{Quantile}_{\beta}\left(L_{n+1}(\hat{\lambda}_{\beta})\right) \leq \alpha \Longleftrightarrow \mathbb{E}\left[\mathbf{1}\left\{L_{n+1}(\hat{\lambda}_{\beta}) > \alpha\right\}\right] \leq 1 - \beta. \tag{6.44}$$

As a consequence, quantile risk control is equivalent to expected risk control (Eq. (6.4)) with loss function $\tilde{L}_i(\lambda) = \mathbf{1}\left\{L_i(\lambda) > \alpha\right\}$. Let

$$\hat{\lambda}_{\beta} = \inf\left\{\lambda \in \Lambda : \frac{1}{n+1}\sum_{i=1}^{n}\mathbf{1}\left\{L_i(\lambda) > \alpha\right\} + \frac{1}{n+1} \leq 1 - \beta\right\}.$$

Proposition 6.5.3. *In the setting of Theorem 6.3.1, Eq. (6.43) is achieved.*

Proof. See Appendix F.1.7. □

Snell et al. (2022) considers the high-probability control of a wider class of quantile-based risks, including the conditional value-at-risk (CVaR). Whether those more general risks can be controlled unconditionally is an open problem for future work.

6.5.3 Controlling multiple risks

Let $L_i(\lambda; \gamma)$ be a family of loss functions indexed by $\gamma \in \Gamma$ for some domain Γ that may have infinitely many elements. A researcher may want to control $\mathbb{E}[L_i(\lambda; \gamma)]$ at level $\alpha(\gamma)$. Equivalently, we need to find an $\hat{\lambda}$ based on $(X_i, Y_i)_{i=1}^n$ such that

$$\sup_{\gamma \in \Gamma} \mathbb{E}\left[\frac{L_i(\hat{\lambda}; \gamma)}{\alpha(\gamma)}\right] \leq 1. \tag{6.45}$$

Though the above worst-case risk is not an expectation, it can still be controlled. Towards this end, we define

$$\hat{\lambda} = \sup_{\gamma \in \Gamma} \hat{\lambda}_\gamma, \text{ where } \hat{\lambda}_\gamma = \inf\left\{\lambda : \frac{1}{n+1} \sum_{i=1}^n L_i(\lambda; \gamma) + \frac{B}{n+1} \leq \alpha(\gamma)\right\}. \tag{6.46}$$

Then the risk is controlled.

Proposition 6.5.4. *In the setting of Theorem 6.3.1, Eq. (6.45) is satisfied.*

Proof. See Appendix F.1.8. □

6.5.4 Adversarial risks

We next show how to control risks defined by adversarial perturbations. We adopt the same notation as Section 6.5.3. Bates et al. (2020) (Section 6.3) discusses the adversarial risk where Γ parametrizes a class of perturbations of X_{n+1}, e.g., $L_i(\lambda; \gamma) = L(X_i + \gamma, Y_i)$ and $\Gamma = \{\gamma : \|\gamma\|_\infty \leq \epsilon\}$. A researcher may want to find a value of $\hat{\lambda}$ based on $(X_i, Y_i)_{i=1}^n$ such that the worst-case risk within this class is controlled,

$$\mathbb{E}\left[\sup_{\gamma \in \Gamma} L_i(\hat{\lambda}; \gamma)\right] \leq \alpha. \tag{6.47}$$

This can be recast as a conformal risk control problem by taking $\tilde{L}_i(\lambda) = \sup_{\gamma \in \Gamma} L_i(\lambda; \gamma)$. Then, the following choice of λ leads to risk control:

$$\hat{\lambda} = \inf\left\{\lambda : \frac{1}{n+1}\sum_{i=1}^{n}\tilde{L}_i(\lambda) + \frac{B}{n+1} \le \alpha\right\}. \tag{6.48}$$

Proposition 6.5.5. *In the setting of Theorem 6.3.1, Eq. (6.47) is satisfied.*

Proof. See Appendix F.1.9. □

6.5.5 U-risk control

For ranking and metric learning, Bates et al. (2020) considered loss functions that depend on two test points. In general, for any $k > 1$ and subset $\mathcal{S} \subset \{1, \ldots, n+k\}$ with $|\mathcal{S}| = k$, let $L_{\mathcal{S}}(\lambda)$ be a loss function.

Our goal is to find $\hat{\lambda}_k$ based on $(X_i, Y_i)_{i=1}^n$ such that

$$\mathbb{E}\left[L_{\{n+1,\ldots,n+k\}}(\hat{\lambda}_k)\right] \le \alpha. \tag{6.49}$$

We call the LHS a U-risk since, for any fixed $\hat{\lambda}_k$, it is the expectation of an order-k U-statistic. As a natural extension, we can define

$$\hat{\lambda}_k = \inf\left\{\lambda : \frac{k!n!}{(n+k)!}\sum_{\substack{\mathcal{S}\subset\{1,\ldots,n\}\\ |\mathcal{S}|=k}} L_{\mathcal{S}}(\lambda) + B\left(1 - \frac{(n!)^2}{(n+k)!(n-k)!}\right) \le \alpha\right\}. \tag{6.50}$$

Again, we define $\hat{\lambda}_k = \lambda_{\max}$ when the right-hand side is an empty set. Then we can prove the following result.

Proposition 6.5.6. *Assume that $L_{\mathcal{S}}(\lambda)$ is non-increasing in λ, right-continuous, and*

$$L_{\mathcal{S}}(\lambda_{\max}) \le \alpha, \quad \sup_{\lambda} L_{\mathcal{S}}(\lambda) \le B < \infty \text{ almost surely.}$$

Then Eq. (6.49) is achieved.

Proof. See Appendix F.1.10. □

6.6 Conclusion

This generalization of conformal prediction broadens its scope to new applications, as shown in Section 6.4. In particular, conformal risk control provides a new perspective to many of the tools developed in earlier chapters, as many of those tools can also be cast in terms of controlling expectations of different risks, even though their respective theoretical analyses are different. The mathematical tools developed in Section 6.3, Section 6.5, and the Appendix may be of independent technical interest, since they provide a new and more general language for studying conformal prediction along with new results about its validity. Important questions remain, such as (1) whether a conformal-type algorithm exists for providing a finite-sample bound when the losses are not monotone but the risk is, (2) whether other functions, such as the value-at-risk, can be controlled with a similar algorithm, and (3) whether there exists a full conformal or cross conformal version of conformal risk control.

7

Conclusion

The contributions put forth in this thesis build towards addressing some of the new challenges associated with deploying large-scale deep learning models in the real world. In particular, in this thesis we focused on two key aspects: efficient adaptive computation in large neural networks and rigorous, general-purpose uncertainty estimation with guarantees. Together, these new tools provide multiple pieces of the puzzle for effectively using today's powerful modern systems in efficient and reliable ways.

Still, there are many open problems in this area, which we briefly explore.

Communicating uncertainty to users

One remaining challenge is in how to best communicate uncertainty to users. Expressing uncertainty in generative language models is a good example. Language modeling provides a flexible framework for solving complex tasks with a unified natural language input and output format. Pre-training also relaxes the need for task-specific data collection and optimization. When combined with their impressive performance on almost every widely-used NLP task considered in the field today, pre-trained large language models (LLMs) have unsurprisingly become extremely popular. Still, like other modeling paradigms, LLMs are susceptible to errors. Uncertainty estimation in generative language modeling, however, can be difficult. For instance, the space of possible model utterances is (in general) infinite, human languages naturally have areas of variation and ambiguity, and syntactic, semantic, and pragmatic uncertainties can be intertwined. Furthermore, while various mathematical formalisms exist for describing and quantifying uncertainty, how to best *communicate* that uncertainty to users (in an audience-aware manner) via language remains quite overlooked.

Efficient continual language learning

In this thesis, we focused on the inference time efficiency of modern models. Our techniques, however, do not apply during training—which is also a computationally expensive procedure. Nevertheless, taking a different angle, the unique shared structure and pre-training strategies of LLMs allow for the same type of models to be used across many different NLP tasks. A typical approach is to take the pre-trained LLM, and then independently fine-tune it on any downstream task that may be encountered. Rather than repeat this (often expensive) procedure over and over again, a natural question to ask is if we can leverage past (supervised or not) tasks as part of a *continual learning* framework in order to make training the (n + 1)th task more efficient (in terms of performance as a function of compute) than if it were the first. Straightforward applications of continual learning, however, can result in *worse* performance over time, whether due to artifacts from non-stationary optimization, or from interference from past, competing and/or contradictory task examples. Developing robust methods that can decide to leverage past data with provably low notions of regret (e.g., that the odds of doing worse than always learning tasks independently is controlled) could lead to more reliable, and efficient, training.

Reliable performance under realistic distribution shifts

A key assumption in our conformal methods is that of data exchangeability. While models may appear to perform well in the lab on datasets where such assumptions hold, or even initially on real data, distribution shifts—that we can expect to happen in nearly any practical scenario—may eventually harm model performance. These shifts may be gradual and subtle (e.g., language usage changing from year to year) or sudden and drastic (e.g., a new MRI machine with different settings is installed in a clinic). Identifying when distribution shifts that adversely affect model performance occur is relevant in situations in which a user cannot easily verify predictions themselves. Moreover, in areas such as predictive medicine, where models make predictions about events that may happen years in the future (e.g., lung cancer risk assessment) immediate corrective feedback is unavailable. A possible solution is *selective prediction*, where a model can choose to abstain from answering if a distribution shift is detected. Attempting to guard against all possible distribution shifts, however, can result in conservative, and unhelpful, model behavior. One possible direction is to develop unsupervised systems for detecting when models require re-training, re-calibration, or absolute abstention, subject to limits on model degradation.

Balancing uncertainty and creativity during drug discovery

The presence of uncertainty is not always a bad thing—in fact it is sometimes a necessary component during discovery and exploration. For example, in-silico screening for drug discovery uses computational tools to discover new drug-like molecules with desireable predicted properties. An inherent point of tension in in-silico screening, however, is that while the most "valuable" molecules are typically those that are distinctly different from those already "discovered" in the training set, these "novel" molecules are also the ones for which accurate predictions—including reliable uncertainty estimates—are the hardest to make. This is compounded by the fact that, in molecular settings, even relatively small changes in structure can result in dramatic differences in functionality (i.e., *activity cliffs*). A possible compromise is to leverage conformal prediction to provide formal confidence intervals for the expected utilities of queried examples, which can then be used as part of a robust portfolio optimization algorithm. Similar tools might be able to handle settings with multiple objectives, which then might be leveraged to construct Pareto optimal sets with high probability.

Summary

In conclusion, this thesis has made significant strides in addressing the challenges of large-scale deep learning models. Our contributions in efficient computation and uncertainty estimation provide practical tools and insights for improving the reliability and efficiency of deep learning models in real-world applications. However, there are still numerous opportunities for future research. By continuing to explore these avenues, we can advance the field of artificial intelligence and develop systems that are not only powerful, but also reliable, trustworthy, and beneficial to society.

Bibliography

Jonathan Alvarsson, Staffan Arvidsson McShane, Ulf Norinder, and Ola Spjuth. Predicting with confidence: Using conformal prediction in drug discovery. *Journal of Pharmaceutical Sciences*, 110(1):42–49, 2021.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *ArXiv preprint arXiv:1606.06565*, 2016.

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then Test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021a.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv 2208.02814*, 2022a. In submission.

Anastasios N Angelopoulos, Amit Pal Kohli, Stephen Bates, Michael Jordan, Jitendra Malik, Thayer Alshaabi, Srigokul Upadhyayula, and Yaniv Romano. Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. In *International Conference on Machine Learning*, pages 717–730. PMLR, 2022b.

Anastasios N Angelopoulos, Karl Krauth, Stephen Bates, Yixin Wang, and Michael I Jordan. Recommendation systems with distribution-free reliability guarantees. *arXiv preprint arXiv:2207.01609*, 2022c.

Anastasios Nikolas Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2021. URL https://arxiv.org/abs/2107.07511.

Anastasios Nikolas Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I. Jordan. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations (ICLR)*, 2021b.

Martin Anthony and John Shawe-Taylor. A result of vapnik with applications. *Discrete Applied Mathematics*, 47(3):207–217, 1993.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. Binarybert: Pushing the limit of bert quantization. *arXiv preprint arXiv:2012.15701*, 2020.

Aaron Baier-Reinio and Hans De Sterck. N-ode transformer: A depth-adaptive variant of the transformer using neural ordinary differential equations. *ArXiv*, abs/2010.11358, 2020.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. Few-shot text classification with distributional signatures. In *International Conference on Learning Representations (ICLR)*, 2020.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. Controlling computation versus quality for neural sequence models. 2020. doi: 10.48550/ARXIV.2002.07106. URL https://arxiv.org/abs/2002.07106.

Rina Barber, Emmanuel Candès, Aaditya Ramdas, and Ryan Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference*, 10, 08 2020. doi: 10.1093/imaiai/iaaa017.

Rina Foygel Barber. Is distribution-free inference possible for binary regression? *Electronic Journal of Statistics*, 14(2):3487 – 3524, 2020.

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *arXiv preprint arXiv:2202.13415*, 2022.

Stephen Bates, Anastasios Nikolas Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution free, risk controlling prediction sets. *ArXiv preprint: 2101.02703*, 2020.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for outliers with conformal p-values. *ArXiv preprint: 2104.08279*, 2021.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 1995.

Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR)*, 2019.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Evgeny Burnaev and Vladimir Vovk. Efficiency of conformalized ridge regression. In *Conference on Learning Theory (COLT)*, 2014.

Berkant Barla Cambazoglu, Hugo Zaragoza, Olivier Chapelle, Jiang Chen, Ciya Liao, Zhaohui Zheng, and Jon Degenhardt. Early exit optimizations for additive machine learned ranking systems. In *WSDM '10*, 2010.

Emmanuel Candès, Lihua Lei, and Zhimei Ren. Conformalized survival analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(1): 24–45, 2023.

L. Carlsson, M. Eklund, and U. Norinder. Aggregated conformal prediction. In *AIAI Workshops*, 2014.

Lars Carlsson, Ernst Ahlberg, Henrik Boström, Ulf Johansson, and Henrik Linusson. Modifications to p-values of conformal predictors. In *Statistical Learning and Data Sciences*, 2015.

Maxime Cauchois, Suyash Gupta, and John C. Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *Journal of Machine Learning Research (JMLR)*, 2021.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001.

Eugene Charniak, Mark Johnson, Micha Elsner, Joseph Austerweil, David Ellis, Isaac Haxton, Catherine Hill, R. Shrivaths, Jeremy Moore, Michael Pozar, and Theresa Vu. Multilevel coarse-to-fine PCFG parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 168–175, New York City, USA, June 2006. Association for Computational Linguistics. URL https://aclanthology.org/N06-1022.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171.

Wenyu Chen, Zhaokai Wang, Wooseok Ha, and Rina Foygel Barber. Trimmed conformal prediction for high-dimensional models. *ArXiv preprint: 1611.09933*, 2016.

Wenyu Chen, Kelli-Jean Chun, and Rina Foygel Barber. Discretized conformal prediction for efficient distribution-free inference. *Stat*, 7(1):e173, 2018.

Victor Chernozhukov, Kaspar Wuthrich, and Yinchu Zhu. Distributional conformal prediction. *ArXiv preprint: 1909.07889*, 2019.

Giovanni Cherubin. Majority vote ensembles of conformal predictors. *Machine Learning*, 108(3):475–488, 2019.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 12 1934. ISSN 0006-3444.

James Davidson and Robert M De Jong. The functional central limit theorem and weak convergence to stochastic integrals ii: fractionally integrated processes. *Econometric Theory*, 16(5):643–666, 2000.

A. P. Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.

Claudio De Stefano, Carlo Sansone, and Mario Vento. To reject or not to reject: That is the question - an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30:84 – 94, 03 2000.

John DeHardt. Generalizations of the Glivenko-Cantelli theorem. *The Annals of Mathematical Statistics*, 42(6):2050–2055, 1971.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations (ICLR)*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Yuntian Deng and Alexander M. Rush. Cascaded text generation with markov transformers. *arXiv preprint: arXiv 2006.01112*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts. 2021.

Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293), 1961.

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator. *The Annals of Mathematical Statistics*, 27(3):642 – 669, 1956. doi: 10.1214/aoms/1177728174. URL https://doi.org/10.1214/aoms/1177728174.

Harrison Edwards and Amos Storkey. Towards a neural statistician. In *International Conference on Learning Representations (ICLR)*, 2017.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research (JMLR)*, 11, 2010.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations (ICLR)*, 2020a. URL https://openreview.net/forum?id=SylO2yStDr.

Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation.

In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 263–273, 2020b. doi: 10.1007/978-3-030-59725-2_26.

Clara Fannjiang, Stephen Bates, Anastasios N Angelopoulos, Jennifer Listgarten, and Michael I Jordan. Conformal prediction for the design problem. *arXiv preprint arXiv:2202.03613*, 2022.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, Jul. 2010.

Michael Figurnov, Artem Sobolev, and Dmitry P. Vetrov. Probabilistic adaptive computation time. *ArXiv*, abs/1712.00386, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. In *International Conference on Learning Representations (ICLR)*, 2021a.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Efficient Conformal Prediction via Cascaded Inference with Expanded Admission. In *ICLR*, 2021b.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning (ICML)*, 2021c.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning (ICML)*, 2021d. URL https://arxiv.org/abs/2102.08898.

Adam Fisch, Tal Schuster, Tommi S. Jaakkola, and Regina Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. In *International Conference on Learning Representations*, 2021e. URL https://openreview.net/forum?id=tnSo6VRLmT.

Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Conformal prediction sets with limited false positives. *arXiv preprint arXiv:2202.07650*, 2022.

Francois Fleuret and Donald Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41(1–2):85–107, 2001.

Dean P. Foster and Rakesh V. Vohra. Asymptotic calibration. *Biometrika*, 85(2): 379–390, 1998.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, 2016.

Alexander Gammerman and Vladimir Vovk. Hedging Predictions in Machine Learning: The Second Computer Journal Lecture . *The Computer Journal*, 50(2), 2007.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Shijie Geng, Peng Gao, Zuohui Fu, and Yongfeng Zhang. Romebert: Robust training of multi-exit bert. 2021.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B*, 69(2): 243–268, 2007.

David L. Gold, Jeffrey C. Miecznikowski, and Song Liu. Error control variability in pathway-based microarray analysis. *Bioinformatics*, 25(17):2216–2221, 2009.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

Alex Graves. Adaptive computation time for recurrent neural networks. 2016.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *ArXiv*, abs/1410.5401, 2014.

Leying Guan. Conformal prediction with localization. 2020.

Antonio Gulli. Ag's corpus of news articles, 2004. URL http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017a.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017b.

Hongyu Guo. A deep network with visual text composition behavior. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 372–377, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2059. URL https://aclanthology.org/P17-2059.

Chirag Gupta and Aaditya Ramdas. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WqoBaaPHS-.

Chirag Gupta, Arun K. Kuchibhotla, and Aaditya K. Ramdas. Nested conformal prediction and quantile out-of-bag ensemble methods. *arXiv: 1910.10562*, 2019.

Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In *International Conference on Machine Learning (ICML)*, 2018.

Martin E. Hellman. The nearest neighbor classification rule with a reject option. *IEEE Transactions on Systems Science and Cybernetics*, 6:179–185, 1970.

Radu Herbei and Marten H. Wegkamp. Classification with reject option. *Canadian Journal of Statistics*, 34(4):709–721, 2006.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning (ICML)*, 2015.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Lior Hirschfeld, Kyle Swanson, Kevin Yang, Regina Barzilay, and Connor W. Coley. Uncertainty quantification using neural networks for molecular property prediction. *ArXiv preprint: 2005.10036*, 2020.

Peter Hoff. Bayes-optimal prediction with frequentist coverage control. *ArXiv preprint: 2105.14045*, 2021.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793, 2020.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Weinberger. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=Hk2aImxAb.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, March 1991.

Yacine Jernite, Edouard Grave, Armand Joulin, and Tomas Mikolov. Variable computation in recurrent neural networks. *arXiv preprint arXiv:1611.06188*, 2016.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5541–5552, 2018.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274, Mar-Apr 2012.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

Ulf Johansson, Ernst Ahlberg, Henrik Boström, Lars Carlsson, Henrik Linusson, and Cecilia Sönströd. Handling small calibration sets in mondrian inductive conformal regressors. In *Statistical Learning and Data Sciences*, 2015.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, USA, 1st edition, 2000. ISBN 0130950696.

Daniel Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York, 2011. ISBN 9780374275631 0374275637. URL https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl_it_dp_o_pdT1_nS_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDFL7.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International Conference on Machine Learning (ICML)*, volume 97, pages 3301–3310. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/kaya19a.html.

Gyuwan Kim and Kyunghyun Cho. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6501–6511, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.508. URL https://aclanthology.org/2021.acl-long.508.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL http://arxiv.org/abs/1412.6980.

Danijel Kivaranovic, Kory D. Johnson, and Hannes Leeb. Adaptive, distribution-free prediction intervals for deep networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3577–3599, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.304. URL https://aclanthology.org/2021.findings-emnlp.304.

Volodymyr Kuleshov and Percy Liang. Calibrated structured prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning (ICML)*, 2018.

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing (NeurIPS)*, 2019.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. ISSN 0036-8075.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*, 2020.

John Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6(10):273–306, 2005. URL http://jmlr.org/papers/v6/langford05a.html.

Kenton Lee, Tom Kwiatkowski, Ankur P. Parikh, and Dipanjan Das. Learning recurrent span representations for extractive question answering. *arXv preprint: 1611.01436*, 2016.

E. L. Lehmann and Joseph P. Romano. Generalizations of the familywise error rate. *Annals of Statistics*, 33(3), 2005a.

E. L. Lehmann and Joseph P. Romano. *Testing statistical hypotheses*. Springer, New York, 2005b. ISBN 0-387-98864-5.

Jing Lei. Classification with confidence. *Biometrika*, 101(4):755–769, 10 2014.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

Lihua Lei and Emmanuel Candès. Conformal inference of counterfactuals and individual treatment effects. 2020. URL https://arxiv.org/abs/2006.06138.

Lihua Lei and Emmanuel J. Candès. Conformal inference of counterfactuals and individual treatment effects. 2020.

Tao Lei. When attention meets fast recurrence: Training language models with reduced compute. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7633–7648, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.602. URL https://aclanthology.org/2021.emnlp-main.602.

Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Lei Li, Yankai Lin, Deli Chen, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. Cascadebert: Accelerating inference of pre-trained language models via calibrated complete models cascade. *arXiv preprint arXiv:2012.14682*, 2020.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

Henrik Linusson, Ulf Johansson, Henrik Boström, and Tuve Löfström. Efficiency comparison of unstable transductive and inductive conformal classifiers. In *Artificial Intelligence Applications and Innovations*, 2014.

Henrik Linusson, Ulf Norinder, Henrik Boström, Ulf Johansson, and Tuve Löfström. On the calibration of aggregated conformal predictors. In *Proceedings of the Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, 2017.

Wei Liu. Multiple tests of a non-hierarchical finite family of hypotheses. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(2):455–461, 1996.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.537. URL https://aclanthology.org/2020.acl-main.537.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. Fastbert: a self-distilling bert with adaptive inference time. *arXiv preprint arXiv:2004.02178*, 2020b.

Xiangyang Liu, Tianxiang Sun, Junliang He, Lingling Wu, Xinyu Zhang, Hao Jiang, Zhao Cao, Xuanjing Huang, and Xipeng Qiu. Towards efficient nlp: A standard evaluation and a strong baseline. *arXiv preprint arXiv:2110.07038*, 2021a.

Yijin Liu, Fandong Meng, Jie Zhou, Yufeng Chen, and Jinan Xu. Faster depth-adaptive transformers. In *AAAI*, 2021b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. 2019.

Zhuang Liu, Zhiqiu Xu, Hung-Ju Wang, Trevor Darrell, and Evan Shelhamer. Anytime dense prediction with confidence adaptivity. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=kNKFOXleuC.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015.

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Joerg Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical Science*, 9(24), 2018.

Peter G. Mikhael, Jeremy Wohlwend, Ludvig Karstens Adam Yala, Justin Xiang, Angelo K. Takigami, Patrick P. Bourgouin, PuiYee Chan, Sofiane Mrah, Wael Amayri, Yu-Hsiang Juan, Cheng-Ta Yang, Yung-Liang Wan, Gigin Lin, Lecia V. Sequist, Florian J. Fintelmann, and Regina Barzilay. Sybil: A validated deep learning model to predict future lung cancer risk from a single low-dose chest computed tomography. *Journal of Clinical Oncology*, 2023. doi: 10.1200/JCO.22.01345.

Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavaš, Shafiq Joty, Alex Wang, and Thomas Wolf, editors. *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.sustainlp-1.0.

Allan H. Murphy and Edward S. Epstein. Verification of probabilistic predictions: A brief review. *Journal of Applied Meteorology and Climatology*, 6(5):748 – 755, 1967.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996. ISBN 0387947248.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *International Conference on Machine Learning (ICML)*, 2005.

Yixin Nie, Haonan Chen, and Mohit Bansal. Combining fact extraction and verification with neural semantic matching networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Ilia Nouretdinov, Thomas Melluish, and Volodya Vovk. Ridge regression confidence machine. In *International Conference on Machine Learning (ICML)*, 2001.

Harris Papadopoulos. Inductive conformal prediction: Theory and application to neural networks. In *Tools in Artificial Intelligence*, chapter 18. IntechOpen, Rijeka, 2008.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*, pages 345–356. Springer, 2002.

Sangdon Park, Osbert Bastani, Nikolai Matni, and Insup Lee. PAC confidence sets for deep neural networks via calibrated prediction. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=BJxVI04YvB.

Sangdon Park, Edgar Dobriban, Insup Lee, and Osbert Bastani. PAC prediction sets for meta-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, November 2016.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, 2010.

Tal Ridnik, Hussam Lawen, Asaf Noy, and Itamar Friedman. TResNet: High performance GPU-dedicated architecture. *arXiv:2003.13630*, 2020.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50:742–54, 05 2010.

Joseph Romano and Michael Wolf. Control of generalized error rates in multiple testing. *Annals of Statistics*, 35(4), 2007.

Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, pages 3543–3553. 2019a. URL https://proceedings.neurips.cc/paper/2019/file/5103c3584b063c431bd1268e9b5e76fb-Paper.pdf.

Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019b.

Yaniv Romano, Rina Foygel Barber, Chiara Sabatti, and Emmanuel Candès. With malice toward none: Assessing uncertainty via equalized coverage. *Harvard Data Science Review*, 4 2020a.

Yaniv Romano, Matteo Sesia, and Emmanuel Candès. Classification with valid and adaptive coverage. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3581–3591. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper/2020/file/244edd7e85dc81602b7615cd705545f5-Paper.pdf.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing (NeurIPS)*, 2020c.

Alexander Rush and Slav Petrov. Vine pruning for efficient multi-pass dependency parsing. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 498–507, Montréal, Canada, June 2012. Association for Computational Linguistics. URL https://aclanthology.org/N12-1054.

Einar Andreas Rødland. Simes' procedure is 'valid on average'. *Biometrika*, 93(3): 742–746, 2006.

Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of American Statistical Association*, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. 2020.

Swami Sankaranarayanan, Anastasios N Angelopoulos, Stephen Bates, Yaniv Romano, and Phillip Isola. Semantic uncertainty intervals for disentangled latent spaces. *arXiv preprint arXiv:2207.10074*, 2022.

Benjamin Sapp, Alexander Toshev, and Ben Taskar. Cascaded models for articulated pose estimation. In *European Conference on Computer Vision (ECCV)*, 2010.

Sanat K. Sarkar and Chung-Kuei Chang. The simes method for multiple hypothesis testing with positively dependent test statistics. *Journal of the American Statistical Association*, 92(440):1601–1608, 1997.

John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Wainwright, Diogo Almeida, Steph Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, Nick Ryder, Alex Paino, Qiming Yuan, Clemens Winter, Ben Wang, Mo Bavarian, Igor Babuschkin, Szymon Sidor, Ingmar Kanitscheider, Mikhail Pavlov, Matthias Plappert, Nik Tezak, Heewoo Jun, William Zhuk, Vitchyr Pong, Lukasz Kaiser, Jerry Tworek, Andrew Carr, Lilian Weng, Sandhini Agarwal, Karl Cobbe, Vineet Kosaraju, Alethea Power, Stanislas Polu, Jesse Han, Raul Puri, Shawn Jain, Benjamin Chess, Christian Gibson, Oleg Boiko, Emy Parparita, Amin Tootoonchian, Kyle Kosic, and Christopher Hesse. Chatgpt. https://openai.com/blog/chatgpt/, 2023.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. In *Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Tal Schuster, Adam Fisch, and Regina Barzilay. Get your vitamin c! robust fact verification with contrastive evidence. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021a. URL https://arxiv.org/abs/2103.08541.

Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021b.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=uLYc4L3C81A.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai. *Commun. ACM*, 63(12):54–63, nov 2020a. doi: 10.1145/3381831. URL https://doi.org/10.1145/3381831.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. The right tool for the job: Matching model and instance complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6640–6651, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.593. URL https://aclanthology.org/2020.acl-main.593.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research (JMLR)*, 9:371–421, June 2008.

Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview. *arXiv preprint: arXiv 2006.06138*, 2020.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821, 2020.

R. J. Simes. An improved Bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3):751–754, 12 1986.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jake C Snell, Thomas P Zollo, Zhun Deng, Toniann Pitassi, and Richard Zemel. Quantile risk control: A flexible framework for bounding the probability of high-loss predictions. *arXiv preprint arXiv:2212.13629*, 2022.

Emil Spjøtvoll. On the Optimality of Some Multiple Comparison Procedures. *The Annals of Mathematical Statistics*, 43(2):398 – 411, 1972.

Samuel Stanton, Wesley Maddox, and Andrew Gordon Wilson. Bayesian optimization with conformal prediction sets. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 959–986. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/stanton23a.html.

Asa Cooper Stickland and Iain Murray. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning*, pages 5986–5995. PMLR, 2019.

Jonathan Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina Donghia, Craig MacNair, Shawn French, Lindsey Carfrae, Zohar Bloom-Ackerman, Victoria Tran, Anush Chiappino-Pepe, Ahmed Badran, Ian Andrews, Emma Chory, George Church, Eric Brown, Tommi Jaakkola, Regina Barzilay, and James Collins. A deep learning approach to antibiotic discovery. *Cell*, 180:688–702.e13, 2020.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

Tianxiang Sun, Xiangyang Liu, Wei Zhu, Zhichao Geng, Lingling Wu, Yilong He, Yuan Ni, Guotong Xie, Xuanjing Huang, and Xipeng Qiu. A simple hash-based early exiting approach for language understanding and generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2409–2421, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.189. URL https://aclanthology.org/2022.findings-acl.189.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020.

Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys (CSUR)*, 2022.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2464–2469, 2016.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. 2017.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, volume 32, pages 2530–2540. 2019. URL https://proceedings.neurips.cc/paper/2019/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://aclanthology.org/W03-0419.

Paolo Toccaceli and Alexander Gammerman. Combination of conformal predictors for classification. In *Sixth Workshop on Conformal and Probabilistic Prediction and Applications*, 2017.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International Conference on Machine Learning (ICML)*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017a.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*. 2017b.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Proceedings of the Asian Conference on Machine Learning*, 2012.

Vladimir Vovk and Ivan Petej. Venn-abers predictors. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.

Vladimir Vovk, Alexander Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *International Conference on Machine Learning*, pages 444–453, 1999.

Vladimir Vovk, Glenn Shafer, and Ilia Nouretdinov. Self-calibrating probability forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2004.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005.

Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. In *International Symposium on Conformal and Probabilistic Prediction with Applications - Volume 9653*, 2016.

Vladimir Vovk, Ivan Petej, Ilia Nouretdinov, Valery Manokhin, and Alexander Gammerman. Computationally efficient versions of conformal predictive distributions. *Neurocomputing*, 397:292 – 308, 2020.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020a.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *The European Conference on Computer Vision (ECCV)*, September 2018a.

Xin Wang, Fisher Yu, Zi-Yi Dou, and Joseph Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. *ArXiv*, abs/1711.09485, 2018b.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. volume 53. Association for Computing Machinery, 2020b.

Larry Wasserman. *All of statistics : a concise course in statistical inference*. Springer, New York, 2010. ISBN 9781441923226 1441923225. URL http://www.amazon.de/All-Statistics-Statistical-Inference-Springer/dp/1441923225/ref=sr_1_2?ie=UTF8&qid=1356099149&sr=8-2.

David Weiss and Benjamin Taskar. Structured prediction cascades. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TrjbxzRcnf-.

Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. Early exiting BERT for efficient document ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.sustainlp-1.11. URL https://aclanthology.org/2020.sustainlp-1.11.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.204. URL https://aclanthology.org/2020.acl-main.204.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. Deebert: Dynamic early exiting for accelerating bert inference. *arXiv preprint arXiv:2004.12993*, 2020c.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. BERxiT: Early exiting for BERT with better fine-tuning and extension to regression. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 91–104, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.8. URL https://aclanthology.org/2021.eacl-main.8.

Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. A survey on green deep learning. *ArXiv*, abs/2111.05193, 2021.

Adam Yala, Peter G. Mikhael, Fredrik Strand, Gigin Lin, Kevin Smith, Yung-Liang Wan, Leslie Lamb, Kevin Hughes, Constance Lehman, and Regina Barzilay. Toward robust mammography-based models for breast cancer risk. *Science Translational Medicine*, 13(578):eaba4373, 2021. doi: 10.1126/scitranslmed.aba4373. URL https://www.science.org/doi/abs/10.1126/scitranslmed.aba4373.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019.

Hongxu Yin, Arash Vahdat, José Manuel Álvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. Adavit: Adaptive tokens for efficient vision transformer. *ArXiv*, abs/2112.07658, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. In *International Conference on Machine Learning (ICML)*, 2020.

Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multiclass calibration. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22313–22324. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bbc92a647199b832ec90d7cf57074e9e-Paper.pdf.

Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.5. URL https://aclanthology.org/2021.naacl-main.5.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. In H. Larochelle,

M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18330–18341. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/file/d4dd111a4fd973394238aca5c05bebe3-Paper.pdf.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020b.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. 2022. doi: 10.48550/ARXIV.2202.09368. URL https://arxiv.org/abs/2202.09368.

Wei Zhu. Leebert: Learned early exit for bert with cross-level optimization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2968–2980, 2021.

A

Appendix for Chapter 1

A.1 Proofs

A.1.1 Proof of Lemma 1.3.3

Proof. It is straightforward to show that for $P = \texttt{pval}(V_{n+1}, V_{1:n})$,

$$P \leq \epsilon \Longleftrightarrow V_{n+1} \text{ is ranked among the } \lfloor \epsilon \cdot (n+1) \rfloor \text{ largest of } V_1, \ldots, V_{n+1}. \tag{A.1}$$

According to our exchangeability assumption over the $n+1$ variables, the right hand side event occurs with at most probability $\lfloor \epsilon \cdot (n+1) \rfloor / (n+1) \leq \epsilon$. □

A.1.2 Proof of Theorem 1.3.4

Proof. For notational convenience, let random variable V_i be the nonconformity score $\mathcal{S}(X_i, Y_i)$. Since the nonconformity scores V_i are constructed symmetrically, then

$$\begin{aligned} ((X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})) &\overset{d}{=} ((X_{\pi(1)}, Y_{\pi(1)}), \ldots, (X_{\pi(n+1)}, Y_{\pi(n+1)})) \\ \Longleftrightarrow (V_1, \ldots, V_{n+1}) &\overset{d}{=} (V_{\pi(1)}, \ldots, V_{\pi(n+1)}) \end{aligned} \tag{A.2}$$

for all permutations $(\pi(1), \dots \pi(n+1))$. Therefore, if $\{(X_i, Y_i)\}_{i=1}^{n+1}$ are exchangeable, then so too are their nonconformal scores $\{V_i\}_{i=1}^{n+1}$. By the definition, we have

$$Y_{n+1} \notin \mathcal{C}_\epsilon(X_{n+1}) \Longleftrightarrow \texttt{pval}\,(V_{n+1}, V_{1:n}) \leq \epsilon. \tag{A.3}$$

Then, using Lemma 1.3.3, $\mathbb{P}(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})) = 1 - \mathbb{P}(Y_{n+1} \notin \mathcal{C}_\epsilon(X_{n+1})) \geq 1 - \epsilon$. □

A.1.3 Proof of Corollary 1.3.5

We first state the following useful lemma on inflated quantiles.

Lemma A.1.1. *Let* Quantile$(\alpha; F)$ *denote the* α *quantile of distribution* F. *Let* $V_{1:n}$ *denote the empirical distribution over random variables* $\{V_1, \dots, V_n\}$. *Furthermore, assume that* V_i, $i = 1, \dots, n+1$ *are exchangeable. Then for any* $\alpha \in (0,1)$, *we have* $\mathbb{P}\left(V_{n+1} \leq \text{Quantile}(\alpha, V_{1:n} \cup \{\infty\})\right) \geq \alpha$.

Proof. Given support $v_1, \dots, v_n \in \mathbb{R}$ for a discrete distribution F, let $q = \text{Quantile}(\alpha; F)$. Any points $v_i > q$ do not affect this quantile, i.e., if we consider a new distribution $\tilde{F}$ where all points $v_i > q$ are mapped to arbitrary values also larger than q then $\text{Quantile}(\alpha; F) = \text{Quantile}(\alpha; \tilde{F})$. Accordingly, for the exchangeable V_i, we have

$$V_{n+1} > \text{Quantile}(\alpha; V_{1:n} \cup \{\infty\}) \Longleftrightarrow V_{n+1} > \text{Quantile}(\alpha; V_{1:(n+1)}). \tag{A.4}$$

Equivalently, we also have that

$$V_{n+1} \leq \text{Quantile}(\alpha; V_{1:n} \cup \{\infty\}) \Longleftrightarrow V_{n+1} \leq \text{Quantile}(\alpha; V_{1:(n+1)}). \tag{A.5}$$

Given the discrete distribution over the $n+1$ variables V_i, $V_{n+1} \leq \text{Quantile}(\alpha; V_{1:(n+1)})$ implies that V_{n+1} is among the $\lceil \alpha(n+1) \rceil$ smallest of $V_{1:(n+1)}$. By exchangeability, this event occurs with probability at least $\frac{\lceil \alpha(n+1) \rceil}{n+1} \geq \alpha$. □

Proof of Corollary 1.3.5. Let $V_i := \mathcal{N}(X_i, Y_i)$. Y_{n+1} is included in $\mathcal{C}_\epsilon(X_{n+1})$ iff $V_{n+1} \leq$

Quantile$(1-\epsilon; V_{1:n} \cup \{\infty\})$. Applying Lemma A.1.1 then gives the result. □

A.2 Hypothesis testing

A number of chapters in this thesis make use of statistical hypothesis testing, which we briefly give informal background for here. See, e.g., Lehmann and Romano (2005b) or Wasserman (2010) for a more comprehensive introduction. Consider the problem of choosing between two hypotheses: the null hypothesis H_0 and the alternative hypothesis H_1, which is H_0's inverse. We define a hypothesis to be a basic statement or assumption that is tested on the basis of observed data. For example, when flipping a coin with bias p, the null hypothesis (H_0) might be that it is not biased in favor of heads, versus the alternative hypothesis (H_1) that it is:

$$H_0\colon p \leq 0.5. \quad \text{vs.} \quad H_1\colon p > 0.5. \tag{A.6}$$

Given data, such as the outcome of n random flips of the coin, we can then make inferences about whether or not H_0 should be accepted or rejected, such that the probability of *falsely* rejecting H_0 is bounded by some $\epsilon \in [0, 1]$. Here the probability is taken over the draw of data given that the null hypothesis is true, i.e.:

$$\mathbb{P}(\text{reject } H_0 \mid H_0 \text{ is true}) \leq \epsilon. \tag{A.7}$$

Example 1. When paired with conformal prediction, where we want to guarantee that the prediction set $\mathcal{C}_\epsilon(X_{n+1})$ satisfies $\mathbb{P}(Y_{n+1} \in \mathcal{C}_\epsilon(X_{n+1})) \geq \epsilon$, we can use statistical testing to test the null hypothesis $H_0\colon y$ is the correct label for X_{n+1}. If we include every label $y \in \mathcal{Y}$ where H_0 is not rejected, then the probability that the one correct y is included is simply $\mathbb{P}(\text{reject } H_0 \mid H_0 \text{ is true})$, which is $\leq \epsilon$ if Eq. (A.7) holds. □

P-values are a useful tool for accepting or rejecting null hypotheses. For our purposes, we define a valid p-value to be any random variable P (typically a statistic of the randomly drawn data) that has a super-uniform distribution under H_0. That is,

$$\mathbb{P}(P \leq u \mid H_0 \text{ is true}) \leq u, \quad \forall u \in [0,1]. \tag{A.8}$$

Example 2. Let K be the number of heads within the sample of n flips of our coin. We can define a p-value for H_0 based on the binomial CDF. Let S be a binomial random variable with success probability p. Then define the random variable

$$P = F_S(K) = \mathbb{P}(S \leq K) = \sum_{i=0}^{K} \binom{n}{i} p^i (1-p)^{n-i}. \tag{A.9}$$

Under H_0, P is super uniform:

$$\begin{aligned}
\mathbb{P}(P \leq u \mid H_0 \textit{ is true}) &= \mathbb{P}(F_S(K) \leq u \mid H_0 \text{ is true}) \\
&= \mathbb{P}(K \leq F_S^{-1}(u) \mid H_0 \textit{ is true}) \\
&\leq F_S(F_S^{-1}(u)) \\
&= u.
\end{aligned} \tag{A.10}$$

□

When a p-value is super-uniform, this allows us to reject H_0 if we observe it to be less than our desired significance level ϵ, as this only happens with probability at most ϵ under H_0. While there are many types of p-values, in this thesis we mainly use p-values based on ranks within an exchangeable collection: see Lemma 1.3.3.

A.2.1 Multiple hypothesis testing

Oftentimes we might be interested in testing multiple hypotheses simultaneously. Continuing the example from earlier, we might now not only be testing just if one coin is biased, but rather an entire coin collection consisting of m different, independent coins. For any one test, the probability of a false rejection of $H_{0,i}$ (in this case, that coin $i \in \{1, \dots, m\}$ is not biased in favor of heads) is bounded by ϵ. Jointly, however, we have that

$$\begin{aligned}\mathbb{P}(\text{any } H_{0,i} \text{ is rejected} \mid H_{0,1}, \ldots, H_{0,m} \text{ are true}) \\ &= 1 - \mathbb{P}(\text{no } H_{0,i} \text{ is rejected} \mid H_{0,1}, \ldots, H_{0,m} \text{ are true}) \quad \text{(A.11)} \\ &\geq 1 - \epsilon^m\end{aligned}$$

which tends to 1 as m grows. In general, the principle is that as the number of simultaneously considered hypotheses becomes larger, the chance of at least one false rejection can become much higher as well, unless proper corrections are made.

Multiple testing corrections are well-studied in statistics, and several exist (with and without assuming certain dependencies between tests). The simplest one is the Bonferroni correction, which has no additional assumptions, as demonstrated next.

Example 3. Given p-values $P_1, \ldots, P_m$ reject $H_{0,i}$ if $P_i < \frac{\epsilon}{m}$.

It is easy to show that this method (called Bonferroni) controls the false rejection rate at level ϵ using the union bound. Let A be the event that any $H_{0,i}$ is false rejected and A_i be the event that $H_{0,i}$ is falsely rejected. Then

$$\mathbb{P}(A) = \mathbb{P}(\bigcup_{i=1}^{m} A_i) \leq \sum_{i=1}^{m} \mathbb{P}(A_i) \leq \sum_{i=1}^{m} \frac{\epsilon}{m} = \epsilon. \quad \text{(A.12)}$$

□

The downside of this method (and others) is that it can be very conservative. When m is large, it can fail to reject any null hypothesis whatsoever, even if there is strong evidence against individual hypotheses. Additional discussion on multiple testing and multiple testing corrections is given in Appendix C.4.

B

Appendix for Chapter 2

B.1 Proofs

B.1.1 Proof of Proposition 2.3.1

Proof. This result is based on Clopper-Pearson confidence interval for Binomial random variables (Clopper and Pearson, 1934). As the events $\mathbf{1}\{\mathcal{G}(X_i;\boldsymbol{\tau}) = \mathcal{F}(X_i)\}$ are i.i.d., the sum is Binomial. Directly applying a one-sided Clopper-Pearson lower bound on the true success rate, $\mathbb{P}(\mathcal{G}(X_i;\boldsymbol{\tau}) = \mathcal{F}(X_i))$, gives the result. □

B.1.2 Proof of Proposition 2.4.1

Proof. We prove by simple calculation using the property assumed in Eq. (2.6).

$$\begin{aligned}
\mathbb{P}(\mathcal{F}_K(X_{n+1}) = \mathcal{F}(X_{n+1})) &= \mathbb{P}(\min \mathcal{C}_\epsilon^c(X_{n+1}) \in \mathcal{I}^c(X_{n+1})) \\
&\geq \mathbb{P}(\mathcal{C}_\epsilon^c(X_{n+1}) \subseteq \mathcal{I}^c(X_{n+1})) \\
&= \mathbb{P}(\mathcal{I}(X_{n+1}) \subseteq \mathcal{C}_\epsilon(X_{n+1})) \\
&\geq 1 - \epsilon.
\end{aligned}$$

□

B.1.3 Proof of Theorem 2.4.4

Proof. For a given k, let $V_k^{(i)} := \mathcal{M}_k(X_i)$ denote the random meta confidence values used for calibration, and $V_k^{(n+1)} := \mathcal{M}_k(X_{n+1})$ the random test point. For all k, $\mathcal{M}_k$ is trained and evaluated on separate data ($\mathcal{D}_{\text{meta}}$ vs $\mathcal{D}_{\text{cal}} \cup \mathcal{D}_{\text{test}}$), preserving exchangeability. Therefore, as $X_{1:n+1}$ are exchangeable, $V_k^{(1:n+1)}$ are also exchangeable.

Layer k is included in $\mathcal{C}_\epsilon^{\text{ind}}$ iff $V_k^{(n+1)} \leq \text{Quantile}(1-\alpha_k, V_k^{(1:n)} \cup \{\infty\})$. For a given k, this happens with probability $\geq 1-\alpha_k$ by Lemma A.1.1. Taken over all $k \in \mathcal{I}(X_{n+1})$ where $|\mathcal{I}(X_{n+1})|$ is at most $l-1$ (i.e., *all* early layers are inconsistent), we have

$$
\begin{aligned}
\mathbb{P}(\mathcal{I}(X_{n+1}) \subseteq \mathcal{C}_\epsilon^{\text{ind}}(X_{n+1})) &= 1 - \mathbb{P}\Big(\bigcup_{k\in\mathcal{I}} \{k \notin \mathcal{C}_\epsilon^{\text{ind}}(X_{n+1})\}\Big) \\
&\geq 1 - \sum_{k\in\mathcal{I}} \mathbb{P}(k \notin \mathcal{C}_\epsilon^{\text{ind}}(X_{n+1}) \\
&= 1 - \sum_{k\in\mathcal{I}} \alpha_k \\
&\geq 1-\epsilon.
\end{aligned}
$$

The last inequality is given by the constraint $\alpha_k = \omega_k \cdot \epsilon$, where $\sum_{i=1}^{l-1} \omega_i = 1$. □

B.1.4 Proof of Theorem 2.4.6

Proof. By the same argument as Theorem 2.4.4, the meta scores $\mathcal{M}_k(X_i)$ are exchangeable. Since $\mathcal{M}_{\text{max}}$ operates symmetrically across all X_i, $M^{(i)} = \mathcal{M}_{\text{max}}(X_i)$ are also exchangeable. Let $M^{(n+1)}$ denote the maximum meta score across inconsistent layers for the new test point. By Lemma A.1.1, this falls below Quantile$(1-\epsilon, M^{(1:n)} \cup \{\infty\})$ with probability at least $1-\epsilon$. Since $M^{(n+1)}$ reflects the maximum meta score, this entails that the meta scores of all other inconsistent layers $k \in \mathcal{I}(X_{n+1})$ for X_{n+1} will be below Quantile$(1-\epsilon, M^{(1:n)} \cup \{\infty\})$ if $M^{(n+1)}$ is, and thereby be included in $\mathcal{C}_\epsilon^{\text{share}}(X_{n+1})$. This gives the bound in Eq. (2.6). □

Nonconformity measure	IMDB Consist.	IMDB Bound	IMDB Layers	VitaminC Consist.	VitaminC Bound	VitaminC Layers	AG News Consist.	AG News Bound	AG News Layers
$1 - \epsilon = 0.95$:									
SM	95.16	93.74	10.39	94.84	94.04	16.60	95.02	93.75	11.63
Meta	94.96	93.72	9.13	94.93	94.12	15.60	94.86	93.58	9.37
$1 - \epsilon = 0.9$:									
SM	90.22	88.30	7.35	89.85	88.59	14.93	89.72	88.01	8.98
Meta	90.19	88.36	7.13	90.00	88.70	13.67	90.14	88.48	6.85

Table B.1.1: Results (dev) using the naive development set calibration method (see §2.3.3). This method tunes the early exit thresholds to get efficient ϵ-consistent predictions on a development set, but does not guarantee that prediction will be ϵ-consistent on new data. "Consist." measures the empirical consistency on a test set, from which we compute a guaranteed lower bound ("Bound") to 99% confidence. The bound is significantly lower than our target $1 - \epsilon$, and the measured consistency in our experiments also falls slightly bellow $1 - \epsilon$ in some cases.

B.2 Implementation details

We implement our early exit Transformers (§2.3) on top of the Transformers library (Wolf et al., 2020).[1] We set d_e to 32 in our experiments. For each task we fix a pre-trained $\mathcal{F}$ and train the early and meta classifiers. We reuse the same training data that was used for $\mathcal{F}$ and divide it to 70/10/20% portions for $\mathcal{D}_{\text{tune}}, \mathcal{D}_{\text{scale}}$ and $\mathcal{D}_{\text{meta}}$, respectively. For classification tasks, we add the temperature scaling step (Guo et al., 2017b) after the early training to improve the calibration of the softmax. We run the scaling for 100 steps on $\mathcal{D}_{\text{scale}}$ using an Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-3}. For the early and meta training we use the same optimizer as for $\mathcal{F}$. We fix $\mathcal{F}$ rather than train it jointly with the new components of $\mathcal{G}$ to avoid any reduction in $\mathcal{F}$'s performance (Xin et al., 2020b). This also makes our method simple to train over any existing Transformer without having to retrain the whole model which could be very costly. Training all parameters of $\mathcal{G}$ jointly can lead to more efficient inference as the early representations will be better suited for classification (Schwartz et al., 2020b; Geng et al., 2021), but potentially with the cost

[1] As discussed in §2.3, our methods can also be applied to any multilayered model such as BERT (Devlin et al., 2019), GPT (Brown et al., 2020), ResNet (He et al., 2015), and others.

of reducing the accuracy of $\mathcal{F}_l$. In the case of joint training, our CATs will provide consistency guarantees with respect to the jointly-trained $\mathcal{F}_l$.

We implement the conformal calibration process in Python and perform retrospective analysis with different random splits of $\mathcal{D}_{\text{cal}}$ and $\mathcal{D}_{\text{test}}$. For Theorem 2.4.4, we simply use the uniform Bonferroni correction, setting $w_k = \frac{1}{l-1} \quad \forall k$. For the naive development set calibration, we use a shared threshold across all layers in order to reduce the examined solution space in Equation 2.3.

B.3 Additional results

In this section, we provide complementary results for the experiments in Chapter 2. All results, except for sections B.3.4 and B.3.5, are with an Albert-xlarge model as $\mathcal{F}$. We note that the results in these tables are based on the development sets, while the tables presented earlier in Chapter 2 report the test set results.

B.3.1 Naïve development set calibration

For completeness, we evaluate the simple, but naïve, calibration method described in §2.3.3. Recall that in this approach we first tune $\boldsymbol{\tau}$ on a development set, and then bound the resulting $\mathcal{G}$'s accuracy using another held-out calibration split. The bound we get is "static", in the sense that we cannot control what value it gives us by further modifying $\boldsymbol{\tau}$. Consequently, we are not able to guarantee that it will satisfy our performance constraint in Eq. (2.1).

Table B.1.1 gives results for our models when using either the Meta or SM confidence measures (which we threshold with $\boldsymbol{\tau}$). We use half of $\mathcal{D}_{\text{cal}}$ to find the minimal threshold that provides ϵ-consistency. Then, we evaluate the threshold on the second half of $\mathcal{D}_{\text{cal}}$ to get the empirical error. We compute the test set bound on this error with a confidence of $\delta = 10^{-2}$. As expected, the lower bound we compute is often significantly below $1 - \epsilon$, as it reflects the uncertainty that our measured consistency is accurate. Often the measured empirical consistency is also slightly below $1 - \epsilon$. At

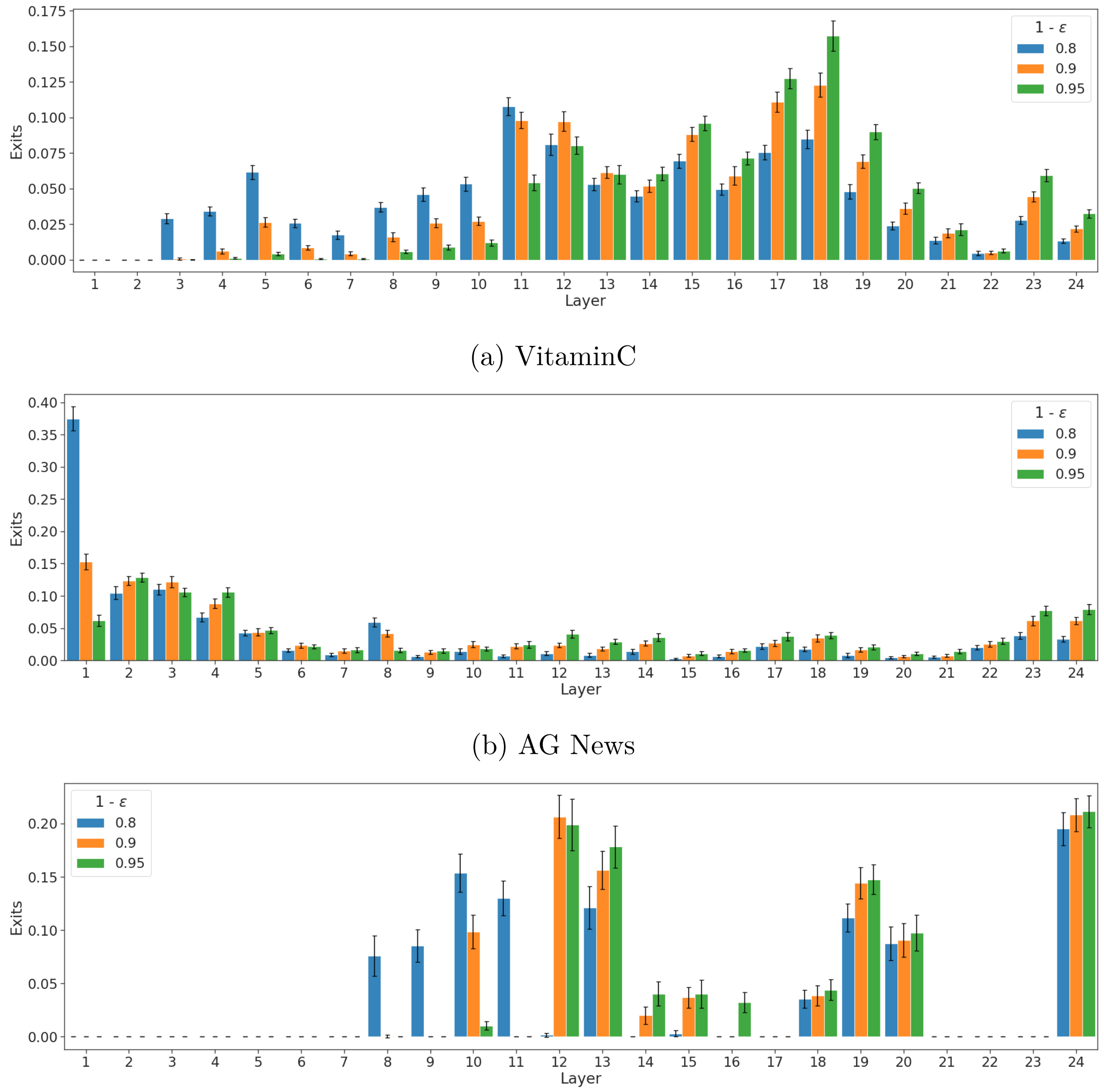
Nonconformity		**IMDB**			**VitaminC**			**AG News**	
measure	Consist.	Acc.	Layers	Consist.	Acc.	Layers	Consist.	Acc.	Layers
$\underline{1 - \epsilon = 0.95}$:		(88.50)			(85.17)			(89.02)	
Random	97.23	91.56	21.57	96.91	87.42	22.71	97.11	91.58	21.60
$D_{\mathrm{KL}}(p_{k-1} \vert\vert p_k)$	97.36	92.49	19.33	96.84	88.85	22.28	97.08	92.46	20.18
$\mathcal{H}(p_k)$	97.28	92.84	12.49	96.79	88.28	17.44	97.15	92.79	14.55
p_k^{diff}	97.28	92.84	12.49	96.83	88.38	17.42	96.96	92.80	12.89
$p_k^{\max}$ (SM)	97.28	92.84	12.49	96.79	88.31	17.40	97.08	92.81	13.23
Meta	96.99	92.24	10.75	96.91	88.29	16.49	96.98	91.98	10.60
$\underline{1 - \epsilon = 0.90}$:		(83.84)			(80.69)			(84.33)	
Random	94.52	89.68	19.21	93.94	85.44	21.47	94.27	89.28	19.01
$D_{\mathrm{KL}}(p_{k-1} \vert\vert p_k)$	94.48	91.36	12.13	93.76	86.81	20.49	93.88	89.98	14.59
$\mathcal{H}(p_k)$	94.49	91.31	9.91	93.67	86.41	16.29	94.54	90.80	13.08
p_k^{diff}	94.49	91.31	9.91	93.67	86.53	16.11	94.02	90.56	10.69
$p_k^{\max}$ (SM)	94.49	91.31	9.91	93.68	86.44	16.13	94.05	90.76	11.01
Meta	94.40	90.45	8.80	93.74	86.17	15.09	94.08	89.72	8.88

Table B.3.1: Results (dev) of our Shared model on the classification tasks using different nonconformity measures. p_k^{diff} and $p_k^{\max}$ are defined in Table 2.1, $D_{\mathrm{KL}}(p_{k-1}||p_k)$ is the Kullback-Leibler Divergence between the previous layer's softmax outputs and the current layer, and $\mathcal{H}(p_k)$ is the entropy of the softmax outputs. Our CP-based Shared method provides the guaranteed consistency with any measure, even random. The benefit, however, of using a better measure is in confidently exiting earlier. Our Meta measure allows the use of least Transformer layers meeting the consistency requirement with enough confidence.

a high level, the overall consistency vs. efficiency trade-off is otherwise broadly similar to the one obtained by the Shared CP calibration.

B.3.2 Nonconformity measure comparison

The test statistic used for a conformal prediction is typically called a nonconformity measure (i.e., for this work this is $\mathcal{M}_k(x)$; in other chapters we have used $\mathcal{N}$ as notation). We experiment with different nonconformity measures as drop-in replacements for $\mathcal{M}_k(x)$, and report the results in Table B.3.1. The conformal calibration guarantees validity with any measure, even a random one, as long as they retain exchangeability. Good measures are ones that are statistically efficient, and will minimize the number of layers required for prediction at the required confidence level. This is a result of smaller $\mathcal{C}_\epsilon$ sets, that tightly cover the inconsistent layers (and hence are more ju-



(a) VitaminC

(b) AG News

(c) STS-B

Figure B.3.1: Distribution of exit layers per tolerance level ϵ (dev sets) with our Shared/ Meta Albert-xlarge model. See Figure 2-5 for IMDB.

dicious with the complement, $\mathcal{C}_{\epsilon}^{c}$). To be consistent with previous work where softmax metrics are used (such as Schwartz et al., 2020b), we use $p_k^{\max}$ as our non-Meta baseline in the main paper. In some settings, however, p_k^{diff} performs slightly better.

B.3.3 Exit layer statistics

Figure B.3.1 depicts the distribution of exit layers for the different tasks with three reference tolerance levels. Reducing ϵ requires greater confidence before exiting, re-

Method	**Amortized time** $(100 \cdot T_{\mathcal{G}}/T_{\mathcal{F}})$			
	IMDB	VitaminC	AG News	STS-B
Thres./ SM	85.56	102.12	112.52	N/A
Thres./ Meta	99.85	109.93	91.95	107.44
Indep./ Meta	89.25	109.57	114.66	130.36
Shared/ SM	67.22	**90.41**	69.99	N/A
Shared/ Meta	**63.99** %	94.97 %	**60.56** %	**99.38** %

Table B.3.2: Complementary results for Table 2.5 with $1 - \epsilon = 0.95$.

Method	**MACs reduction** $(\lvert\mathcal{F}\rvert/\lvert\mathcal{G}\rvert)$			
	IMDB	VitaminC	AG News	STS-B
Thres./ SM	1.45	1.20	1.08	N/A
Thres./ Meta	1.35	1.22	1.48	1.25
Indep./ Meta	1.53	1.22	1.17	1.02
Shared/ SM	1.90	1.37	1.81	N/A
Shared/ Meta	×**2.22**	×**1.43**	×**2.36**	×**1.36**

Table B.3.3: Complementary results for Table 2.6 with $1 - \epsilon = 0.95$.

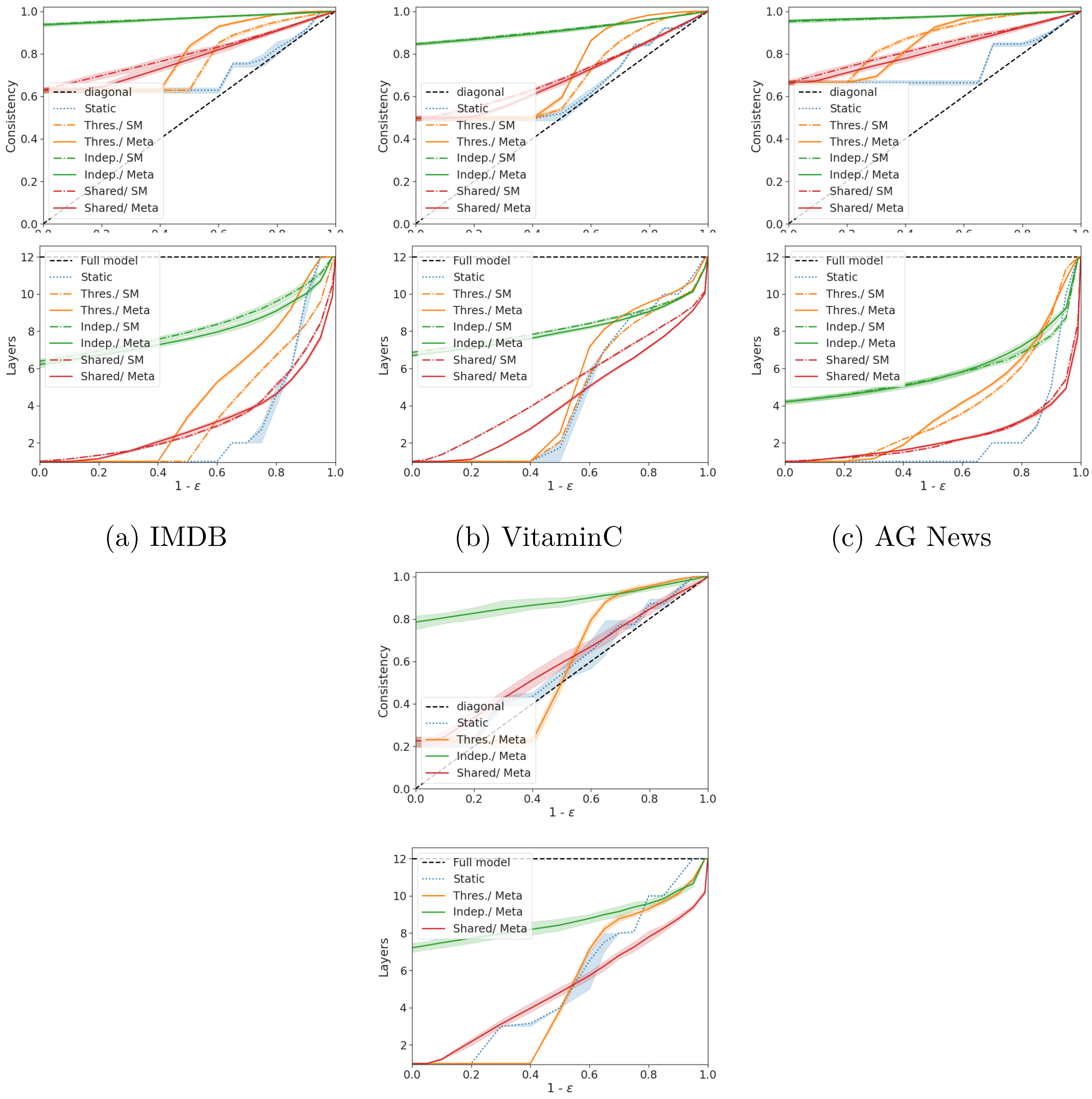
sulting in later exits on average. We provide example inputs with their respective exit layer in Appendix B.4.

B.3.4 Albert-base results

Figure B.3.2 reports the classification and regression results with an Albert-base 12-layers model. The trends are similar to the larger 24-layers version. Again, we see the efficacy of our Shared conformal calibration and the Meta nonconformity scores. For example, the AG News CAT Shared/ Meta model can preserve 95% consistency while using less than 5 Transformer layers on average.

B.3.5 RoBERTa-large results

Figure B.3.3 shows the results of our methods on top of the RoBERTa-large 24-layers Transformer. One main difference between RoBERTa and Albert, is that Albert shares the same parameters across all layers, essentially applying the same function recursively, whereas RoBERTa learns different parameters per layer. Yet, our method is agnostic to such differences and, as observed in the plots, results in similar trends.



(a) IMDB

(b) VitaminC

(c) AG News

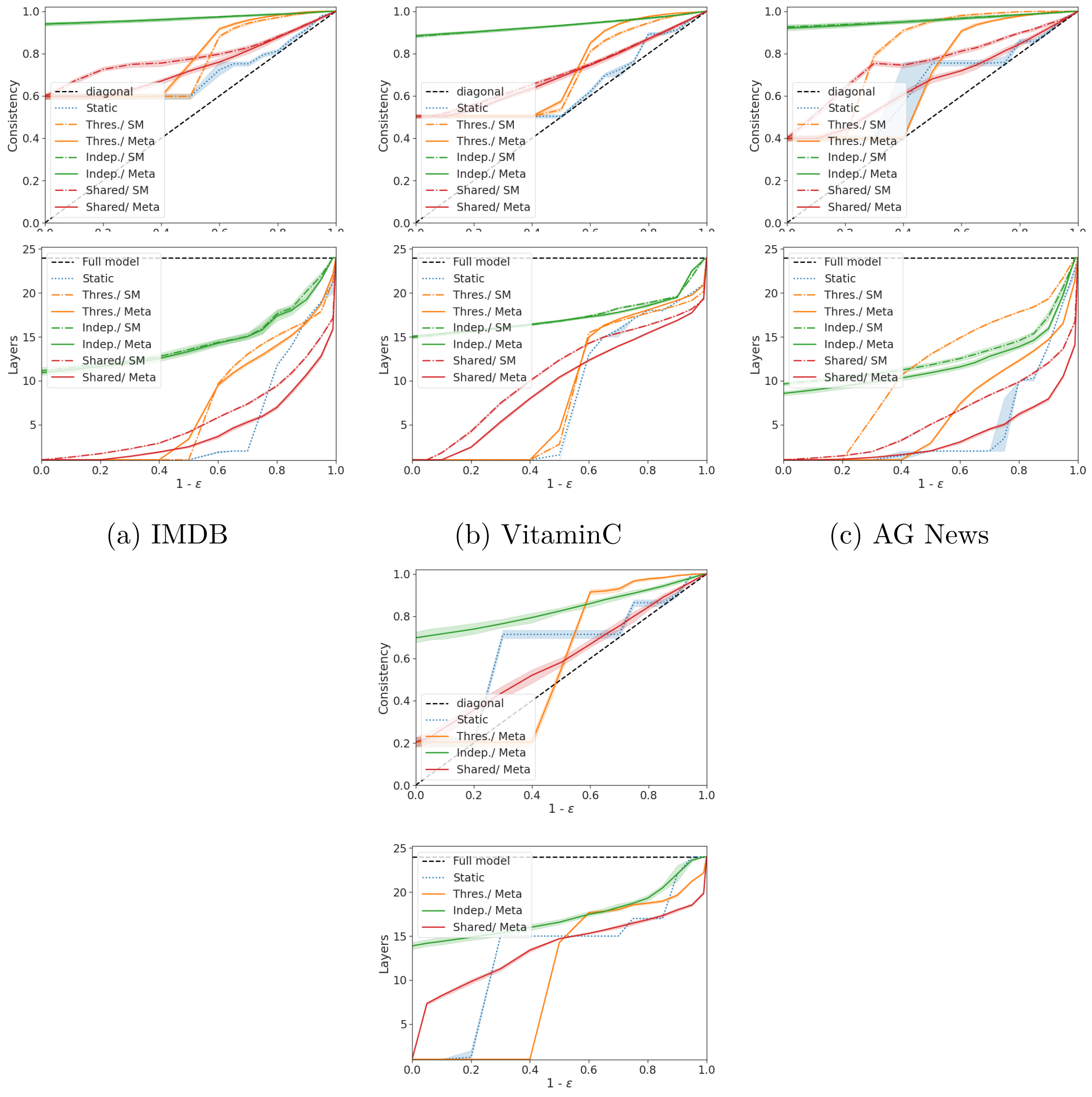
(d) STS-B

Figure B.3.2: Development set results with an Albert-base 12-layers model as $\mathcal{F}$.

The value of our Meta classifier compared to the softmax response is even greater with the RoBERTa model.

B.4 Example predictions

Tables B.4.1-B.4.4 report examples of inputs for different tasks and the number of layers that our Albert-xlarge CAT with $\epsilon = 0.1$ required. These examples suggest



(a) IMDB

(b) VitaminC

(c) AG News

(d) STS-B

Figure B.3.3: Development set results with an RoBERTa-large 24-layers model as $\mathcal{F}$.

that “easier” inputs (e.g., containing cue phrases or having large overlaps in sentence-pair tasks) might require less layers. In contrast, more complicated inputs (e.g., using less common language or requiring numerical analysis) can lead to additional computational effort until the desired confidence is obtained.

IMDB (Maas et al., 2011)

Exit layer	Gold label	Input
1	Pos	Without question, film is a powerful medium, more so now than ever before, due to the accessibility of DVD/video, which gives the filmmaker the added assurance that his story or message is going to be seen by possibly millions of people. [...]
4	Neg	This movie was obscenely obvious and predictable. The scenes were poorly written and acted even worse.
10	Pos	I think Gerard's comments on the doc hit the nail on the head. Interesting film, but very long. [...]
15	Pos	here in Germany it was only shown on TV one time. today, as everything becomes mainstream, it's absolute impossible, to watch a film like this again on the screen. maybe it's the same in USA [...]
20	Neg	I tried to be patient and open-minded but found myself in a coma-like state. I wish I would have brought my duck and goose feather pillow... [...]
24	Neg	Hypothetical situations abound, one-time director Harry Ralston gives us the ultimate post-apocalyptic glimpse with the world dead, left in the streets, in the stores, and throughout the landscape, sans in the middle of a forgotten desert. [...]

Table B.4.1: Number of Transformer layers used for example inputs from the IMBD test set with our Shared/Meta CAT with a tolerance level of $\epsilon = 0.1$

VitaminC (Schuster et al., 2021a)

Exit layer	Gold label	Input
3	Sup	Claim: Another movie titled The SpongeBob Movie: Sponge on the Run is scheduled for release in 2020. Evidence: A second film titled The SpongeBob Movie : Sponge Out of Water was released in 2015, and another titled The SpongeBob Movie: Sponge on the Run is scheduled for release in 2020.
5	Sup	Claim: Julie Bishop offered a defence of her nation's intelligence cooperation with America. Evidence: The Australian Foreign Minister Julie Bishop stated that the acts of Edward Snowden were treachery and offered a staunch defence of her nation's intelligence co-operation with America.
10	NEI	Claim: The character Leslie hurts her head on the window in the film 10 Cloverfield Lane. Evidence: Michelle realizes Howard was right and returns his keys.
15	Sup	Claim: Halakha laws are independent of being physically present in the Land of Israel. Evidence: The codification efforts that culminated in the Shulchan Aruch divide the law into four sections, including only laws that do not depend on being physically present in the Land of Israel.
20	Sup	Claim: Germany has recorded less than 74,510 cases of coronavirus , including under 830 deaths. Evidence: 74,508 cases have been reported with 821 deaths and approximately 16,100 recoveries.
24	NEI	Claim: For the 2015-16 school year , the undergraduate fee at USF is under $43,000. Evidence: Undergraduate tuition at USF is $44,040 for the 2016-17 school year.

Table B.4.2: Number of Transformer layers used for example inputs from the VitaminC test set with our Shared/Meta CAT with a tolerance level of $\epsilon = 0.1$

AG News (Gulli, 2004; Zhang et al., 2015)

Exit layer	Gold label	Input
1	Business	Crude Oil Rises on Speculation Cold Weather May Increase Demand Crude oil futures are headed for their biggest weekly gain in 21 months [...]
5	Sports	NHL Owner Is Criticized for Talking of Replacement Players The day before the regular season was supposed to open [...]
15	World	Scotch Whisky eyes Asian and Eastern European markets (AFP) AFP - A favourite tipple among connoisseurs the world over, whisky is treated with almost religious reverence on the Hebridean [...]
20	Business	Arthritis drug withdrawn after trial A prescription painkiller used by more than 250,000 Australians to treat arthritis has been withdrawn from sale after a clinical trial found it doubled the risk [...]
24	Sci/Tech	Airbus drops out of Microsoft appeal Aircraft builder withdraws its request to intervene in Microsoft's antitrust appeal; Boeing also forgoes intervention.

Table B.4.3: Number of Transformer layers used for example inputs from the AGNews test set with our Shared/Meta CAT with a tolerance level of $\epsilon = 0.1$

STS-B (Cer et al., 2017)

Exit layer	Gold label	Input
10	0.6	Sent. 1: A child wearing blue and white shorts is jumping in the surf. Sent. 2: A girl wearing green twists something in her hands.
15	2.8	Sent. 1: Saudi Arabia gets a seat at the UN Security Council Sent. 2: Saudi Arabia rejects seat on UN Security Council
20	4.2	Sent. 1: a small bird sitting on a branch in winter. Sent. 2: A small bird perched on an icy branch.
24	3.0	Sent. 1: It depends entirely on your company and your contract. Sent. 2: It depends on your company.

Table B.4.4: Number of Transformer layers used for example inputs from the STS-B test set with our Shared/Meta CAT with a tolerance level of $\epsilon = 0.1$

C

Appendix for Chapter 3

C.1 Proofs

C.1.1 Proof of Theorem 3.3.4

Proof. We begin by establishing exchangeability of the minimal nonconformity scores, $\mathcal{N}_g^{\min}(X_i, Y_i)$, as defined in Eq. (3.4). We simplify notation once again by letting $V_i := \mathcal{N}(X_i, Y_i)$, and $M_i := \mathcal{N}_g^{\min}(X_i, Y_i)$. Since (X_i, Y_i) are exchangeable and $\mathcal{N}$ is fixed, V_i are also exchangeable. Then, since the label expansion and subsequent min operator are taken *point-wise* over the V_i, we retain symmetry, and therefore M_i are also exchangeable. Next, we want to prove that $\mathbb{P}\left(|\mathcal{A}_g(X_{n+1}, Y_{n+1}) \cap \mathcal{C}_\epsilon^{\min}(X_{n+1})| \geq 1\right) \geq 1 - \epsilon$. Let $|\mathcal{A}_g(X_{n+1}, Y_{n+1})| = k$, where $\mathcal{A}_g(X_{n+1}, Y_{n+1}) = \{\bar{Y}_{n+1}^1, \ldots, \bar{Y}_{n+1}^k\}$. Then:

$$
\begin{aligned}
&\mathbb{P}\left(\left|\mathcal{A}_g(X_{n+1}, Y_{n+1}) \cap \mathcal{C}_\epsilon^{\min}\right| \geq 1\right) = \mathbb{P}\left(\bigcup_{i=1}^{k} \bar{Y}_i \in \mathcal{C}_\epsilon^{\min}\right) \\
&\geq \max\left\{\mathbb{P}\left(\bar{Y}_1 \in \mathcal{C}_\epsilon^{\min}\right), \ldots, \mathbb{P}\left(\bar{Y}_k \in \mathcal{C}_\epsilon^{\min}\right)\right\} \\
&= \max\left\{\mathbb{P}\left(\texttt{pval}(\mathcal{N}(X_{n+1}, \bar{Y}_{n+1}^1), M_{1:n}) > \epsilon\right), \ldots, \mathbb{P}\left(\texttt{pval}(\mathcal{N}(X_{n+1}, \bar{Y}_{n+1}^k), M_{1:n}) > \epsilon\right)\right\} \\
&\overset{(i)}{=} \mathbb{P}\left(\,\texttt{pval}(M_{n+1}, M_{1:n}) > \epsilon\,\right) \\
&\overset{(ii)}{\geq} 1 - \epsilon
\end{aligned}
\tag{C.1}
$$

where (i) is by construction of M_{n+1} (assuming unique non-conformity scores or tie-breaking in `pval`), and (ii) comes from applying Lemma 1.3.3.

We now prove the second part of Theorem 3.3.4: that $\mathbb{E}\left[|\mathcal{C}_\epsilon^{\min}(X_{n+1})|\right] \leq \mathbb{E}\left[|\mathcal{C}_\epsilon(X_{n+1})|\right]$. Let $\mathbf{1}\{y_i \in \mathcal{C}_\epsilon\}$ be the indicator random variable that $y_i \in \mathcal{Y}$ is included in $\mathcal{C}_\epsilon$. Then:

$$\begin{aligned}
\mathbb{E}\left[|\mathcal{C}_\epsilon|\right] &= \mathbb{E}\left[\sum_{i=1}^{|\mathcal{Y}|} \mathbf{1}\{y_i \in \mathcal{C}_\epsilon\}\right] \\
&= \sum_{i=1}^{|\mathcal{Y}|} \mathbb{E}[\mathbf{1}\{y_i \in \mathcal{C}_\epsilon\}] \\
&= \sum_{i=1}^{|\mathcal{Y}|} \mathbb{P}\left(\texttt{pval}(\mathcal{N}(X_{n+1}, y_i), V_{1:n}) > \epsilon\right).
\end{aligned} \tag{C.2}$$

By the same derivation, we have

$$\mathbb{E}\left[|\mathcal{C}_\epsilon^{\min}|\right] = \sum_{i=1}^{|\mathcal{Y}|} \mathbb{P}\left(\texttt{pval}(\mathcal{N}(X_{n+1}, y_i), M_{1:n}) > \epsilon\right). \tag{C.3}$$

Since $M_i \leq V_i,\ \forall i \in [1, n+1]$, we have $V_{n+1} \leq M_i \implies V_{n+1} \leq V_i$, and it is the easy to see from the definition of the `pval` that this also implies

$$\mathbb{P}(\texttt{pval}(\mathcal{N}(X_{n+1}, y), M_{1:n}) > \epsilon) \leq \mathbb{P}(\texttt{pval}(\mathcal{N}(X_{n+1}, y), V_{1:n}) > \epsilon), \quad \forall y \in \mathcal{Y}. \tag{C.4}$$

It follows that $\mathbb{E}\left[|\mathcal{C}_\epsilon^{\min}|\right] \leq \mathbb{E}\left[|\mathcal{C}_\epsilon|\right]$. □

C.1.2 Proof of Theorem 3.4.1

Proof. We restate our assumption that $\mathcal{M}$ is a valid multiple hypothesis testing correction procedure that properly controls the family-wise error rate,

$$\tilde{P} = \mathcal{M}\left(P_1, \ldots, P_m\right) \text{ s.t. } \mathbb{P}\left(\tilde{P} \leq \epsilon \mid Y \text{ is correct}\right) \leq \epsilon, \tag{C.5}$$

and that it is element-wise monotonic

$$\left(P_1, \ldots, P_m\right) \preccurlyeq \left(\hat{P}_1, \ldots, \hat{P}_m\right) \Longrightarrow \mathcal{M}\left(P_1, \ldots, P_m\right) \leq \mathcal{M}\left(\hat{P}_1, \ldots, \hat{P}_m\right), \tag{C.6}$$

where P_i are p-values produced by different nonconformity measures for the same point (X, Y), and $\preccurlyeq$ operates element-wise.[1] First we show that all $\mathcal{C}_\epsilon^m$ constructed via Eq. (3.11) satisfies Eq. (3.1) when all p-values are known and no pruning has been done. $\mathcal{M}$ operates element-wise over examples, therefore exchangeability is conserved and all basic individual p-value computations from nonconformity scores are valid as before. Then, by the construction of $\mathcal{C}_\epsilon^m(X_{n+1})$, we have

$$Y_{n+1} \notin \mathcal{C}_\epsilon(X_{n+1}) \iff \tilde{P} \leq \epsilon, \tag{C.7}$$

and

$$\begin{aligned} \mathbb{P}\left(Y_{n+1} \in \mathcal{C}_\epsilon^m(X_{n+1})\right) &= 1 - \mathbb{P}\left(Y_{n+1} \notin \mathcal{C}_\epsilon^m(X_{n+1})\right) \\ &= 1 - \mathbb{P}\left(\tilde{P} \leq \epsilon\right) \geq 1 - \epsilon, \end{aligned} \tag{C.8}$$

where the final inequality comes from the first assumption on $\mathcal{M}$. Next, we want to prove that early pruning does not remove any candidates y that would *not* be removed from $\mathcal{C}_\epsilon^m$. When all p-values after step j are not yet known, we set $P_{k>j}$ to 1. Using the element-wise monotonicity of $\mathcal{M}$, we have

$$\mathcal{M}\left(P_1, \ldots, P_j, 1, \ldots 1\right) \geq \mathcal{M}\left(P_1, \ldots, P_m\right) \tag{C.9}$$

Therefore,

$$\left\{y \in \mathcal{Y} \colon \tilde{P}_j^{(y)} > \epsilon\right\} \supseteq \left\{y \in \mathcal{Y} \colon \tilde{P}_m^{(y)} > \epsilon\right\}, \tag{C.10}$$

yielding $y \in \mathcal{C}_\epsilon^m(X_{n+1}) \Rightarrow y \in \mathcal{C}_\epsilon^j(X_{n+1}), \forall y \in \mathcal{Y}$. We finish by using the earlier result for $\mathcal{C}_\epsilon^m(X_{n+1})$ to get

$$\mathbb{P}\left(Y_{n+1} \in \mathcal{C}_\epsilon^j(X_{n+1})\right) \geq \mathbb{P}\left(Y_{n+1} \in \mathcal{C}_\epsilon^m(X_{n+1})\right) \geq 1 - \epsilon, \quad \forall j \in [1, m]. \tag{C.11}$$

□

[1] As previously noted, though we formally require this, we are unaware of any common $\mathcal{M}$ satisfying Eq. (3.9) but not Eq. (3.10). See Appendix C.4 for common MHT corrections.

C.2 Implementation details

Multiple answers. When multiple answers are given (i.e., y is a set) we take $\mathcal{A}_g(x, y)$ as the union of all the admissible answers, along with any additional answers expanded by g. For the standard conformal prediction baseline, we calibrate on one of the answers at a time, chosen uniformly at random. Note that this is important to preserve equivalent sample sizes across *min*-calibrated CP and standard CP.[2]

Open-Domain QA. We use the open-domain setting of the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019). In this task, the goal is to find a short span from any article in Wikipedia that answers the given question. Questions in the NQ dataset were sourced from real Google search queries, and human annotators identified answer spans to the queries in Wikipedia articles (we use the short answer span setting, and only consider answerable, non-boolean questions).

We use the open-source, pre-trained DPR model for retrieval (i.e., the document retriever) and the BERT model for question answering (i.e., the document reader) provided by Karpukhin et al. (2020). To summarize briefly, the DPR model is trained to maximize the dot-product similarity between the dense representations (obtained via BERT embeddings) of the question and the passage that contains the answer. For candidate passages, we use the Wikipedia-based corpus processed by Karpukhin et al. (2020), where each article is split into disjoint passages of 100 words, resulting in a total of 21,015,324 passages. The DPR model pre-computes dense representations for all passages and indexes them with FAISS (Johnson et al., 2019) for efficient retrieval. At test time, relevant documents for a given dense question encoding are retrieved via fast similarity search. The reader model is a standard BERT model with an independent span prediction head. This model encodes the question and passage jointly, and therefore, the representations cannot be pre-computed. For each token, the model outputs independent scores for being the start or end of the answer span. We also follow Karpukhin et al. (2020) by using the output of the "`[CLS]`" token

[2] Another possibility would be to calibrate on *all* the given answers, but we found this does worse.

Task	Metric	Description
QA	retriever	$-1\cdot$ sim. score of the question/paragraph by the retrieval model.
	passage	$-1\cdot$ logit of the passage selection score by BERT.
	span start	$-1\cdot$ logit of the answer's first token by BERT.
	span end	$-1\cdot$ logit of the answer's last token by BERT.
	span sum	span start + span end.
IR	BM25	$-1\cdot$ BM25 similarity score between query and candidate.
	CLS logit	$-1\cdot$ logit of the claim-evidence pair by ALBERT.
DR	RF	$-1\cdot$ score of the candidate by the Random Forest model.
	MPNN	$-1\cdot$ score of the candidate by the directed MPNN.

Table C.2.1: Nonconformity measures used in our experiments.

to get a passage selection score from the reader model (to augment the score of the retriever). For ease of experimentation, we only consider the top 5000 answer spans per question—taken as the top 100 passages (ranked by the retriever) and the top 50 spans per passage (ranked by the reader). In order to be able to evaluate all $\epsilon \in (0, 1)$ we discard questions whose answers do not fall within this selection. We retain 6750/8757 questions from the validation set and 2895/3610 from the test set.

We compose the cascade for this task using four metrics: the retriever score, followed by the reader's passage selection score, the "span start" score, and the "span end" score. For the non-cascaded models, we take the sum of the span start and span end scores as a single metric. See Table C.2.1.

IR. We use the FEVER dataset for evidence retrieval and fact verification (Thorne et al., 2018). We focus on the retrieval part of this task. Note that the retrieved evidence can then be used to verify the correctness of the claim automatically (Nie et al., 2019; Schuster et al., 2019), or manually by a user. We follow the splits of the Eraser benchmark (DeYoung et al., 2020) that contain 97,957 claims for training, 6,122 claims for validation, and 6,111 claims for test. The evidence needs to be retrieved from a set of 40,133 unique sentences collected from 4,099 total Wikipedia articles.

We compose the cascade using two metrics: an efficient BM25 model, and a neural sentence-pair classifier. Our BM25 retriever uses the default configuration available in the Gensim library (Řehůřek and Sojka, 2010). We perform simple preprocessing to the text, including removing punctuation and adding word stems. We also add the article title to each sentence. Our neural classifier (CLS) is built on top of ALBERT-Base and is trained with BCE on (claim, evidence) pairs. We collect 10 negative pairs for each positive one by randomly selecting other sentences from the same article as the correct evidence. For shorter articles, we extend the negative sampling to also include the top (spurious) candidates indentified by the BM25 retriever.

DR. We construct a molecular property screening task using the ChEMBL dataset (see Mayr et al., 2018). Given a specified constraint such as "*is active for property A and property B but not property C*", we want to retrieve at least one molecule from a given set of candidates that satisfies this constraint. The motivation of this task is to simulate in-silico screening for drug discovery, where it is often the case where chemists will searching for a new molecule that satisfies several constraints (such as toxicity and efficacy limits), out of a pool of many possible molecular candidates.

We split the ChEMBL dataset into a 60-20-20 split of molecules, where 60% of molecules are separated into a train set, 20% into a validation set, and 20% into a test set. Next, we take all properties that have more than 1000 labeled molecules (of which at least 100 are positive and 100 are negative, to avoid highly imbalanced properties). Of these ∼200 properties, we take all N choose K combinations that have at least 100 molecules with all K properties labelled (ChEMBL has many missing values). We set K to 3. For each combination, we randomly sample an assignment for each property (i.e., $\{\text{active}, \text{inactive}\}^K$). We keep 5000 combinations for each of validation and test sets. The molecules for each of the combinations are only sourced from their respective splits (i.e., molecular candidates for constraints in the property combination validation split only come from the molecule validation split). Therefore, at inference time, given a combination we have never seen before, on a molecules we have never seen before, we must try to retrieve at least one molecule that has the desired combination assignment.

Both our random forest (RF) and the directed Message Passing Neural Network (MPNN) were implemented using the `chemprop` repository (Yang et al., 2019). The RF model is based on the Scikit library (Pedregosa et al., 2011) and uses 2048-bit binary Morgan fingerprints (Rogers and Hahn, 2010) of radius 2 to independently predict all target properties (active or inactive) for a given molecule. The RF model is fast to run during inference, even on a single CPU. The MPNN model uses graph convolutions to learn a deep molecular representation, that is shared across property predictions. Each property value (active/inactive) is predicted using an independent classifier head. The final prediction is based on an ensemble of 5 models, trained with different random seeds. Given a combination assignment $(Z_1 = z_1, \ldots, Z_k = z_k)$, for both the RF and MPNN models, we take the nonconformity score as the model's negative log-likelihood, where the likelihood is computed independently, i.e.

$$p_\theta(Z_1 = z_1, \ldots, Z_k = z_k) = \prod p_\theta(Z_i = z_i). \tag{C.12}$$

C.3 Additional results

We provide supplemental experimental results to those shown in §3.6. In C.3.1 we show an example of a *closed-domain* QA task where cascading multiple measures *boosts* the predictive efficiency to the extent that it outweighs the MHT correction factor. In C.3.2 we compare our method to heuristic methods at fixed ϵ.

C.3.1 Complementary conformal cascades for closed-domain QA

The motivation in this work for conformalized cascades is to improve *computational efficiency* by allowing cheaper models to filter the candidate space prior to running more expensive and more powerful models. Though not guaranteed, in some cases it is also possible that combining different nonconformity scores together has an overall synergistic effect that outweighs the generally conservative effects of the MHT corrections. This is similar in theory to ensembles or mixtures-of-experts (Jacobs et al., 1991), and similar results have been reported for combined conformal prediction (Carlsson et al., 2014; Linusson et al., 2017; Toccaceli and Gammerman, 2017).

While the tasks in §3.6 focus on cascades that are designed primarily for efficiency, here we also explore cascades for the smaller-scale task of closed-domain questions answering on the SQuAD 2.0 dataset (Rajpurkar et al., 2018). This task isolates the *document reader* aspect of the open-domain QA pipeline, in which a relevant passage is already given. We cascade two primary models: (1) a span extractor (EXT) that gives independent scores for the start and end positions of the answer span, and (2) a more expensive answer classifier (CLS) that considers the entire candidate span (i.e., models the start and end jointly). We briefly outline the implementation details before giving the results below.

The EXT model uses the ALBERT-Base QA model (Lan et al., 2020). It is trained to maximize the likelihood of the answer span $[i, j]$, where $p(\text{start} = i, \text{end} = j)$ is modeled as $p(\text{start} = i)p(\text{end} = j)$. During inference, the model computes all $\mathcal{O}(n^2)$ start and end position scores, and predicts the pair with the highest sum. We use the start and end position scores as two separate nonconformity measures. The CLS model is also built on top of ALBERT-Base, and is similar to Lee et al. (2016). Instead of scoring start and end positions independently, we concatenate the corresponding hidden representations at tokens i (start) and j (end), and score them jointly using an MLP. We then train with binary cross-entropy (BCE) over correct and incorrect answers. We limit the number of negative samples (incorrect answers) to the top 64 incorrect predictions of the EXT model.

The authors keep the original test set hidden; for ease of CP-specific evaluation we re-split the development set randomly by article. This results in 5,302 questions in our validation set, and 6,571 questions in our test set. The average length of a paragraph in the evaluation set is 127 words. Together with the "no answer" option, a question for a paragraph of that length will have 8,129 candidate answer spans. For the purposes of experimentation, we filter out questions for which the EXT model does not rank the correct answer within the top 150 predictions (so we can compute the full $(0, 1)$ range of ϵ tractably). This discards less than 0.5% of questions.

Our results across several values of epsilon are given in Table C.3.1. As in the tasks in §3.6, the *min*-calibrated CPs improve over the baseline CP. In this case, however, the cascaded CP also consistently outperforms the CP with only a single measure.

$1-\epsilon$	CP		*min* CP		Cascaded *min* CP		
	Succ.	$\lvert\mathcal{C}_\epsilon\rvert$	Succ.	$\lvert\mathcal{C}_\epsilon^{\min}\rvert$	Succ.	$\lvert\mathcal{C}_\epsilon^{\min}\rvert$	Amortized cost
0.99	1.00	17.69	0.99	6.68	0.99	6.31	0.50
0.95	0.98	5.14	0.95	3.14	0.96	2.45	0.42
0.90	0.95	3.09	0.90	1.98	0.92	1.76	0.40
0.80	0.86	1.66	0.80	1.31	0.83	1.24	0.38

Table C.3.1: CP results on the SQuAD 2.0 dataset.

C.3.2 Heuristic methods

In addition to evaluating our improvements over regular conformal prediction, we compare our conformal method to other common heuristics for making set-valued predictions. Specifically, we consider baseline methods that given some scoring function $\texttt{score}(x, y)$ and threshold τ to define the output set of predictions $\mathcal{B}$ at $x \in \mathcal{X}$ as

$$\mathcal{B}(x, \tau) := \{y \in \mathcal{Y} \colon \texttt{score}(x, y) \geq \tau\}, \tag{C.13}$$

where τ is then tuned on the calibration set to find the largest threshold for the desired accuracy:

$$\tau_\epsilon^* := \sup\left\{\tau \colon \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{y_i \in \mathcal{B}(x, \tau)\} \geq 1 - \epsilon\right\}. \tag{C.14}$$

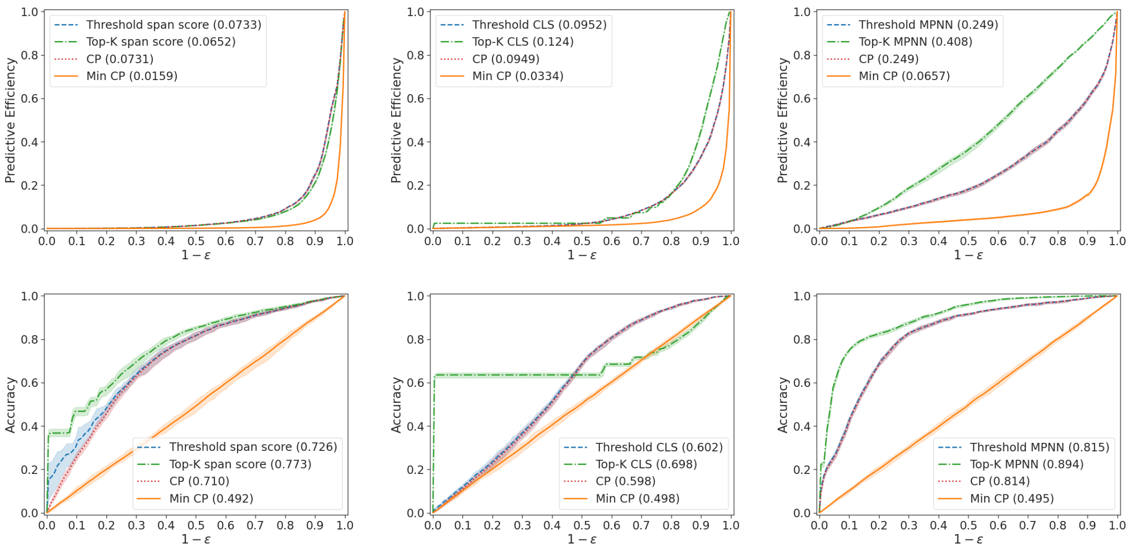
The prediction for the test point x_{n+1} is then $\mathcal{B}(x_{n+1}, \tau_\epsilon^*)$. We consider two variants: (1) fixed top-k, where $\texttt{score} := -\text{rank}(\texttt{metric}(x_{n+1}, y))$ according to some metric, and (2) raw thresholding, where $\texttt{score} := \texttt{metric}(x_{n+1}, y)$, i.e., some raw, unnormalized metric. Top-k is simple and relatively robust to the variance of the metric used, but as it doesn't depend on x, it also means that both easy examples and hard examples are treated the same (giving prediction sets that are too large in the former, and too small in the latter). Raw metric thresholding, on the other hand, gives dynamically-sized prediction sets, but is more sensitive to metric variance. We emphasize that these baselines do not provide theoretical coverage guarantees for potentially non-i.i.d. finite samples.

When ignoring smoothing factors and restricting ourselves to a single, non-cascaded metric, it is straightforward to see that choosing the nonconformity measure to be $\mathcal{S} := \text{rank}(\texttt{metric}(x, y))$ or simply $\mathcal{S} := -\texttt{metric}(x, y)$ makes the set $\mathcal{C}_n(x_{n+1})$ equivalent to $\mathcal{B}(x_{n+1}, \tilde{\tau}_\epsilon)$, where

$$\tilde{\tau}_\epsilon := \text{Quantile}\left(1 - \epsilon, \widehat{F}\left(\{\mathcal{S}(x_1, y_1), \ldots, \mathcal{S}(x_n, y_n)\} \cup \{\infty\}\right)\right). \tag{C.15}$$

We write $\widehat{F}(\mathcal{V})$ to denote the empirical distribution of set $\mathcal{V}$. For large enough n, $\tilde{\tau}_\epsilon$ becomes nearly identical to τ^*_ϵ. Note, however, that this comparison only applies to the *split* (i.e., inductive) conformal prediction setting. Metrics computed without relying on a held-out calibration set (as *full* CP is able to do) must treat the $n+1$ data points symmetrically, a key feature of CP. Our methods extend to multiple cascaded metrics, finite n, and the full CP setting.

We compare predictive efficiency results when using top-k and threshold heuristics versus our CP methods in Figure C.3.1 and Table C.3.2. As expected, with large enough n, the CP and Threshold methods are nearly equivalent. In some cases, top-k outperforms the threshold- and CP-based methods that use raw scores (this is likely due to variance in the densities of high scoring candidates across examples). When optimizing for admissible answers, our *min*-calibrated CP improves over all three methods. Note that optimizing for admissible answers is also applicable for the heuristic methods, in which case the trends will be similar to that of CP.



(a) QA (b) IR (c) DR

Figure C.3.1: We show predictive efficiency and accuracy across various target accuracies (values of $1-\epsilon$). As expected, with a large enough calibration set, the performance of the threshold heuristic is similar to CP. The top-k threshold is harder to tune, resulting in lower than the desired accuracy for some values of IR, and sometimes in unnecessary large prediction sets. Specific values of ϵ are compared in Table C.3.2.

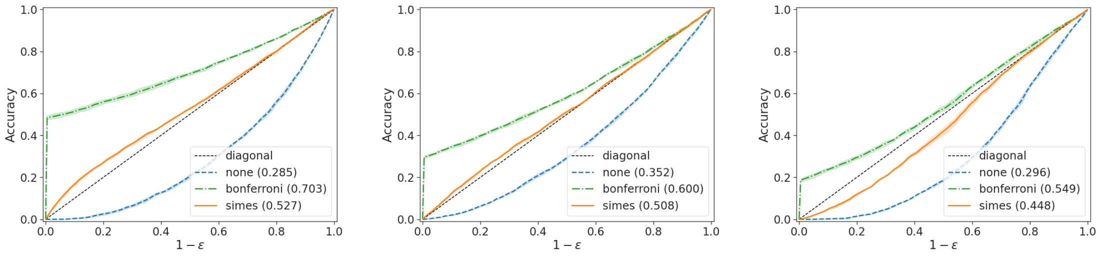
Task	Target Acc.	Threshold		Top-k		Baseline CP		*min*-CP	
		Acc.	$\lvert\mathcal{B}\rvert$	Acc.	$\lvert\mathcal{B}\rvert$	Acc.	$\lvert\mathcal{C}_\epsilon\rvert$	Acc.	$\lvert\mathcal{C}_\epsilon^{\min}\rvert$
QA	0.90	0.98	1243.8	0.98	1041.7	0.98	1245.7	0.90	198.0
	0.80	0.94	452.8	0.95	390.5	0.94	453.5	0.80	58.7
	0.70	0.91	228.8	0.92	203.3	0.91	227.8	0.70	22.1
	0.60	0.87	128.4	0.89	120.5	0.87	127.8	0.60	10.6
IR	0.99	1.00	33.6	1.00	41.0	1.00	33.6	0.99	17.8
	0.95	1.00	20.9	0.95	29.8	1.00	20.9	0.95	7.4
	0.90	0.98	13.8	0.89	18.7	0.98	13.8	0.90	4.0
	0.80	0.95	6.7	0.78	6.7	0.95	6.7	0.80	1.7
DR	0.90	0.99	429.5	1.00	652.9	0.99	429.8	0.90	84.1
	0.80	0.97	305.8	1.00	525.1	0.97	305.8	0.80	42.7
	0.70	0.96	216.8	0.99	398.3	0.96	216.6	0.70	28.1
	0.60	0.94	145.7	0.98	266.5	0.94	145.6	0.60	19.3

Table C.3.2: Non-CP baseline ($\mathcal{B}$) results on the test set for different target accuracy values, compared to CP methods ($\lvert\mathcal{C}_\epsilon\rvert$ and $\lvert\mathcal{C}_\epsilon^{\min}\rvert$).

C.4 Multiple hypothesis testing

As we discuss in §3.4, naively combining multiple hypothesis tests will lead to an increased family-wise error rate (FWER). As a simple analogy, suppose the rejection decision is based on a random toss of a fair coin. Then we will reject the correct label 50% of the time. However, if we toss two coins and reject if *either* one is “heads”, then we will reject the correct label 75% of the time. For a visual example, see the uncorrected cascaded CP (blue dashed lined) in Figure C.4.1. It demonstrates that combining nonconformal measures without using a MHT correction can result in accuracies smaller than $1 - \epsilon$ (i.e., invalid coverage).

Several methods exist for correcting for MHT by bounding the FWER (with different assumptions on the dependencies between the hypothesis tests). We experiment with the Bonferroni and Simes procedures. We test the corrections on each task’s validation set and find Simes to work well for QA and IR, but not to hold for DR (See Figure C.4.1). Therefore, we use the Bonferroni correction for DR and Simes for the other two tasks. For completeness, we briefly describe the two methods and the assumptions they rely on below (extensive additional details can be found in the literature).



(a) QA (b) IR (c) DR

Figure C.4.1: Success rate against tolerance thresholds with different methods for MHT correction (validation set). Not applying any correction leads to a in-valid classifier (the success rate is below the diagonal). The Bonferroni method is conservative, leading to a valid classifier, but sometimes with a higher accuracy rate than necessary (a result of having excessively large $\mathcal{C}_\epsilon$). The Simes correction works for the QA and IR tasks and provides a tighter bound for them. The Simes correction does not work for the DR task, likely due to a violation of its MTP2 assumption, but the Bonferroni method provides a relatively tight correction there—especially for small ϵ.

C.4.1 Bonferroni Procedure

The Bonferroni correction is easy to apply and doesn’t require assumptions on the dependencies between the hypothesis tests used (e.g., independent, positively dependent, *etc*). However, it is generally very conservative. The Bonferroni correction scales ϵ by a factor of $1/m$, and uses the union bound on the p-values $(P_1, \ldots, P_m)$ to get:

$$\mathbb{P}\left(Y_{n+1} \notin \mathcal{C}_{\epsilon}^{m}(X_{n+1})\right) = \mathbb{P}\left(\bigcup_{i=1}^{m} P_i \leq \frac{\epsilon}{m}\right) \leq \sum_{i=1}^{m} \mathbb{P}\left(P_i \leq \frac{\epsilon}{m}\right) = \epsilon. \tag{C.16}$$

We get the same result by scaling each p-value by m, and take the combined p-value to be the minimum of the scaled p-values. It is easy to show that this is monotonic.

C.4.2 Simes Procedure

The Simes procedure (Simes, 1986) allows a stricter bound on the FWER when the measures are multivariate totally positive (Sarkar and Chang, 1997), which usually holds in practice (Rødland, 2006). To apply the correction, we first sort the p-values in ascending order, and then perform an order-dependent correction where the correction factor decreases as the index of the p-value increases. Specifically, if $(P_{(1)}, \ldots P_{(m)})$ are the sorted p-values $(P_1, \ldots P_m)$ in an m-level cascade, we modify the p-values to be $P_{(i)}^{\text{Sim}} = m \cdot \frac{p_{(i)}}{i}$. We take the final combined p-value to be the minimum of the corrected p-values. It is easy to show that this correction is monotonic.

D

Appendix for Chapter 4

D.1 Proofs

D.1.1 Proof of Lemma 4.3.3

Proof. Notice that

$$t \leq t' \implies \{v_j \colon v_j < t\} \subseteq \{v_j \colon v_j < t'\}. \tag{D.1}$$

As an immediate consequence,

$$\begin{aligned} t \leq t' &\implies \{\mathrm{FP}(z, \mathcal{S}_j) \colon v_j \leq t\} \subseteq \{\mathrm{FP}(z, \mathcal{S}_j) \colon v_j \leq t\} && \text{(D.2)} \\ &\implies \max\{\mathrm{FP}(z, \mathcal{S}_j) \colon v_j < t\} \leq \max\{\mathrm{FP}(z, \mathcal{S}_j) \colon v_j < t'\} && \text{(D.3)} \\ &\implies \mathrm{FP}_{\max}(x, z, t) \leq \mathrm{FP}_{\max}(x, z, t'). && \text{(D.4)} \end{aligned}$$

□

D.1.2 Proof of Theorem 4.3.4

Our proof of Theorem 4.3.4 builds on conformal risk control (Angelopoulos et al., 2022a, see also Chapter 6). We restate the main result here:

Theorem D.1.1 (CRC)**.** *Let $L_i \colon \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, n+1$ be exchangeable functions, where $L_i(t)$ is non-increasing in t. Also, take $g \colon \mathbb{R} \to \mathbb{R}$ where $g(x)$ is non-decreasing in x. Further assume that $g \circ L_i$ is right-continuous, and*

$$\inf_t g(L_i(t)) < \gamma, \qquad \sup_t g(L_i(t)) \le B < \infty \text{ almost surely.} \tag{D.5}$$

Then

$$\mathbb{E}[g \circ L_{n+1}(T(\gamma; g))] \le \gamma, \tag{D.6}$$

where

$$T(\gamma; g) = \inf \left\{ t \colon \frac{1}{n+1} \left(B + \sum_{i=1}^{n} g(L_i(t)) \right) \le \gamma \right\}. \tag{D.7}$$

Proof. See Angelopoulos et al. (2022a). □

We provide an additional corollary for lower-bounding function R_{n+1}, where $R_1, \ldots, R_{n+1}$ are now non-decreasing exchangeable functions (as opposed to non-increasing).

Corollary D.1.2 (CRC, lower bound, non-decreasing case)**.** *Similar to the setting in Theorem D.1.1, let $R_i \colon \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, n+1$ be exchangeable functions, where $R_i(t)$ is non-decreasing in t. Also, take $g \colon \mathbb{R} \to \mathbb{R}$ where $g(x)$ is non-decreasing in x. Further assume that $g \circ R_i$ is right-continuous, and*

$$\inf_t g(R_i(t)) \ge 0, \qquad \sup_t g(R_i(t)) > C \ge \gamma \text{ almost surely.} \tag{D.8}$$

For any $\gamma \le 0$, define the random variable $T(\gamma, g)$ as

$$T(\gamma; g) := \inf \left\{ t \colon \frac{1}{n+1} \sum_{i=1}^{n} g(R_i(t)) \ge \gamma \right\}, \tag{D.9}$$

where we define $\inf \varnothing = \infty$*. Then $\mathbb{E}[g \circ R_{n+1}(T(\gamma; g))] \ge \gamma$.*

Proof. Let

$$T'(\gamma; g) := \inf \left\{ t \colon \frac{1}{n+1} \sum_{i=1}^{n+1} g(R_i(t)) \geq \gamma \right\}. \tag{D.10}$$

Since $\inf_t g(R_i(t)) \geq 0$, $\sup_t g(R_i(t)) > C \geq \gamma$, $T'(\gamma; g)$ and $T(\gamma, g)$ are both well-defined almost surely.

Since $\inf_t g(R_i(t)) \geq 0$,

$$\frac{1}{n+1} \sum_{i=1}^{n} g(R_i(t)) \geq \gamma \Longrightarrow \frac{1}{n+1} \sum_{i=1}^{n+1} g(R_i(t)) \geq \gamma. \tag{D.11}$$

Thus, $T'(\gamma; g) \leq T(\gamma; g)$. Since $g(R_i(t))$ is non-decreasing in t,

$$\mathbb{E}[g \circ R_{n+1}(T(\gamma; g))] \geq \mathbb{E}[g \circ R_{n+1}(T'(\gamma; g))]. \tag{D.12}$$

Let E_f be the unordered set (bag) of $\{R_1, \dots, R_{n+1}\}$. Then $T'(\gamma; g)$ is a function of E_f, and is a constant conditional on E_f. Exchangeability of R_i and right-continuity of $g \circ R_i$ imply

$$\mathbb{E}[g \circ R_{n+1}(T'(\gamma; g)) \mid E_f] = \frac{1}{n+1} \sum_{i=1}^{n+1} g \circ R_i(T'(\gamma; g)) \geq \gamma. \tag{D.13}$$

As this is true given any E_f, we can take the expectation over E_f to yield

$$\mathbb{E}[\mathbb{E}[g \circ R_{n+1}(T'(\gamma; g)) \mid E_f]] \geq \gamma. \tag{D.14}$$

The proof is completed by applying Eq. (D.12). □

Proof of Theorem 4.3.4. By Lemma 4.3.3, we have that $\mathrm{FP}_{\max}(x, z, t)$ is non-decreasing in t. It is also easy to verify that $\mathrm{FP}_{\max}(x, z, t)$ is left-continuous in t and preserves exchangeability, so that $\mathrm{FP}_{\max}(X_i, Z_i, t)$ are exchangeable functions of t. Next, define

$$\mathrm{FP}^{-}_{\max}(x, z, t) := \mathrm{FP}_{\max}(x, z, -t), \tag{D.15}$$

so that $\mathrm{FP}^{-}_{\max}(x,z,t)$ is non-increasing in t and right-continuous. Define random variables T'_k and $T'_{k,\delta}$ as

$$T'_k = \inf\left\{t \in \mathbb{R} \colon \frac{B + \sum_{i=1}^{n} \mathrm{FP}^{-}_{\max}(X_i, Z_i, t)}{n+1} \leq k\right\} \quad \text{and} \tag{D.16}$$

$$T'_{k,\delta} = \inf\left\{t \in \mathbb{R} \colon \frac{\sum_{i=1}^{n} \mathbf{1}\{\mathrm{FP}^{-}_{\max}(X_i, Z_i, t) \leq k\}}{n+1} \geq 1-\delta\right\}. \tag{D.17}$$

We then have $T_k = -T'_k$ and $T_{k,\delta} = -T'_{k,\delta}$, which gives

$$\mathbb{E}\Big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\Big] = \mathbb{E}\Big[\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_k)\Big] \quad \text{and} \tag{D.18}$$

$$\mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \leq k\Big) = \mathbb{P}\Big(\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \leq k\Big). \tag{D.19}$$

(Part 1) We first prove $\mathbb{E}\Big[\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_k)\Big] \leq k$.

Since B is finite, we have that $\sup_t \mathrm{FP}^{-}_{\max}(x,z,t) \leq \max_j |\mathcal{S}_j| \leq B < \infty$. As we assume nonconformity scores are finite, we also have $\inf_t \mathrm{FP}^{-}_{\max}(x,z,t) = 0 < k \in \mathbb{R}_{>0}$. Let $L_i(t) = \mathrm{FP}^{-}_{\max}(X_i, Z_i, t)$ and $g(x) = x$. Theorem D.1.1 gives

$$\mathbb{E}\Big[\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_k)\Big] \leq k. \tag{D.20}$$

Substituting Eq. (D.18) gives $\mathbb{E}\Big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\Big] \leq k$.

(Part 2) We now prove $\mathbb{P}\Big(\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \leq k\Big) \geq 1-\delta$.

Let $L_i(t) = \mathbf{1}\{\mathrm{FP}^{-}_{\max}(X_i, Z_i, t) \leq k\}$. Let $g(x) = x$. As shown earlier, $\mathrm{FP}^{-}_{\max}(X_i, Z_i, t)$ is non-increasing, right-continuous; as a result $L_i(t)$ is non-decreasing, right-continuous. Let $\gamma = 1-\delta \in (0,1)$. Since $g(L_i(t)) \in \{0,1\}$ and $V_{i,j}$ are finite, it is easy to see that we have $\sup_t g(L_i(t)) = 1 \geq \gamma$ and $\inf_t g(L_i(t)) = 0 \geq 0$.

Applying Corollary D.1.2 gives

$$\mathbb{E}\Big[\mathbf{1}\{\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \leq k\}\Big] = \mathbb{P}\Big(\mathrm{FP}^{-}_{\max}(X_{n+1}, Z_{n+1}, T'_{k,\delta}) \leq k\Big) \tag{D.21}$$

$$\geq \gamma \tag{D.22}$$

$$= 1 - \delta. \tag{D.23}$$

Substituting Eq. (D.19) gives $\mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \leq k\Big) \geq 1 - \delta$.

□

D.1.3 Proof of Proposition 4.3.8

Proof. We first prove simultaneous validity of candidate sets indexed by $j \in \mathcal{J}$. By definition we have

$$\mathrm{FP}(Z_{n+1}, \mathcal{S}_j) \leq \mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{\circ}) \quad \forall j \in \mathcal{J}, \tag{D.24}$$

which implies

$$\mathbb{E}\Big[\mathrm{FP}(Z_{n+1}, \mathcal{S}_j)\Big] \leq \mathbb{E}\Big[\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_k)\Big] \quad \text{and} \tag{D.25}$$

$$\mathbb{P}\Big(\mathrm{FP}(Z_{n+1}, \mathcal{S}_j) \leq k\Big) \geq \mathbb{P}\Big(\mathrm{FP}_{\max}(X_{n+1}, Z_{n+1}, T_{k,\delta}) \leq k\Big) \tag{D.26}$$

simultaneously $\forall j \in \mathcal{J}$. Theorem 4.3.4 then implies validity.

We now show maximal TPR (a simple outcome). If $\mathcal{S}_j \subseteq \mathcal{S}_{j'}$ then $y_c \in \mathcal{S}_j \implies y_c \in \mathcal{S}_{j'}$ for any $y_c \in z \subseteq \mathcal{Y}$. Therefore

$$\mathcal{S}_j \subseteq \mathcal{S}_{j'} \implies \mathrm{TPP}(z, \mathcal{S}_j) \leq \mathrm{TPP}(z, \mathcal{S}_{j'}). \tag{D.27}$$

Since candidate sets are nested,

$$j \leq j' \implies \mathcal{S}_j \subseteq \mathcal{S}_{j'}, \tag{D.28}$$

and

$$\text{TPP}(z, \mathcal{S}_{\max \mathcal{J}}) \geq \text{TPP}(z, \mathcal{S}_{j'}) \quad \forall j' \in \mathcal{J}. \tag{D.29}$$

Since this is true for all (x, z),

$$\mathbb{E}\Big[\text{TPP}(Z_{n+1}, \mathcal{S}_{\max \mathcal{J}})\Big] = \sup_{h \in \mathcal{H}} \mathbb{E}\Big[\text{TPP}(Z_{n+1}, \mathcal{S}_{h \circ \mathcal{J}})\Big] \tag{D.30}$$

where $\mathcal{H}$ is the space of all possible index selection policies. □

D.2 Additional training details

We provide additional details on $\mathcal{F}$'s training data. First, recall that in our setup we have an input space $\mathcal{X}$ and label space $\mathcal{Y}$. For every input $x \in \mathcal{X}$, there are (assumed to be) potentially multiple true positives $z \subseteq \mathcal{Y}$. Referring to Algorithm 3, during training (TRAIN) we split whatever multi-label training data we have into two sets. The first (larger split) is used to learn a likelihood model (which can either be simply training independent binary predictors, or something with label dependencies that are explicitly accounted for, e.g., with a CRF). The second (smaller split) is used to train $\mathcal{F}$. From this second split of data, again, we have inputs x and sets of true positives z. For every input x, however, we can sample combinatorially many candidate output sets $z \subseteq \mathcal{Y}$, and measure the (ground truth) number of false positives by comparing z' with z. This (x, z) candidate is then used as a training instance for $\mathcal{F}$, where our target is to directly estimate the number of false positives, $|z \setminus z|$.

In practice, we construct candidate sets greedily by ranking output labels individually, and then combining them into nested sets, i.e., $\{y_{\pi(1)}\}, \{y_{\pi(1)}, y_{\pi(2)}\}, \{y_{\pi(1)}, y_{\pi(2)}, y_{\pi(3)}\}$, etc. This is aligned with our eventual calibration and inference time procedure. As with testing, we only train on candidate sequences up to length B (recall that B is used as a hyper-parameter for a cutoff for the maximum number of considered sets—both for efficiency and technical details for ensuring CP guarantees). This also allows us to train $\mathcal{F}$ quite efficiently, since, instead of randomly sampling candidate sets z, we create them greedily (as we would during testing). Concretely, suppose for an input x with true label set z we have the set of labels ranked by individual likelihood, $\{y_{\pi(1)}, y_{\pi(2)}, y_{\pi(3)}, \ldots, y_{\pi(B)}\}$. For each ranked $y_{\pi(i)}$, we also have a corresponding $\{0, 1\}$ target $y'_{\pi(i)}$ that indicates if $y_{\pi(i)}$ is a false positive, or not. Suppose, in this example, this sequence of labels is $\{1, 0, 1, \ldots, 1\}$. Taking the cumulative sum, our total number of FP targets for the batch are then then provided by the set $\{1, 1, 2, \ldots, B - |z|\}$. We train $\mathcal{F}$ to predict each of these targets by pairing the shared input x with each of the nested candidate sets $\{y_{\pi(1)}, \ldots, y_{\pi(k)}\} \subseteq \{y_{\pi(1)}, \ldots, y_{\pi(B)}\}$, paired with the sum of its false positives, $\sum_{i \leq k} y'_{\pi(i)}$. In total, for a training split of n examples, we will have $n \times B$ prediction targets.

D.3 Implementation and dataset details

In-silico screening. We construct a molecular property screening task using the ChEMBL dataset (Mayr et al., 2018). Given a specified constraint such as "*is active for property A but not property B*," we want to retrieve at least one molecule from a given set of candidates that satisfies this constraint. The input for each molecule is its SMILES string, a notational format that specifies its molecular structure. The motivation of this task is to simulate in-silico screening for drug discovery, where it is often the case where chemists are searching for a new molecule that satisfies several constraints (such as toxicity and efficacy limits), out of a pool of many candidates.

We split the ChEMBL dataset into a 60-20-20 split of molecules, where 60% of molecules are separated into a train set, 20% into a validation set, and 20% into a test set. Next, we take all properties that have at least 50 positive and negative examples (to avoid highly imbalanced properties). Of these properties, we take all N choose K combinations that have at least 100 molecules with all K properties labelled (ChEMBL has many missing values). We set K to 2. For each combination, we randomly sample an assignment for each property (i.e., $\{\text{active}, \text{inactive}\}^K$). We discard combinations for which more than 90% of labeled molecules satisfy the constraint. We keep 5000 combinations for testing, 767 for validation, and 4375 for training. The molecules for each of the combinations are only sourced from their respective splits (i.e., molecular candidates for constraints in the property combination validation split only come from the molecule validation split). Therefore, at inference time, given a combination we have never seen before, on a molecules we have never seen before, we must try to retrieve the molecules that have the desired combination assignment.

Our directed Message Passing Neural Network (MPNN) is implemented using the `chemprop`[1] repository (Yang et al., 2019). The MPNN model uses graph convolutions to learn a deep molecular representation, that is shared across property predictions. Each property value (active/inactive) is predicted using an independent classifier head. The final prediction is based on an ensemble of 5 models, trained with different random seeds. Given a combination assignment $(Z_1 = z_1, \ldots, Z_k = z_k)$, we naively compute the joint likelihood independently, i.e.,

$$p_\theta(Z_1 = z_1, \ldots, Z_k = z_k) = \prod p_\theta(Z_i = z_i). \tag{D.31}$$

Object detection. As discussed in §4.4, we use the MS-COCO dataset (Lin et al., 2014) to evaluate our conformal object detection. MS-COCO consists of images of complex everyday scenes containing 80 object categories (such as person, bicycle, dog, car, etc.), multiple of which may be contained in any given example. Since the official test set is hidden, we use the $5k$ examples from the development set and randomly partition them into sets of size $1k$, $1k$, and $3k$ for calibration, validation, and testing, respectively. The EfficientDet model (Tan et al., 2020)[2] for extracting bounding boxes uses a pipeline of three neural networks to extract deep features, and then predict candidates. The model also uses a non-maximum suppression (NMS) post-processing step to reduce the total number of predictions by keeping only the one with the maximum score across highly overlapping prediction boxes. We merge the predictions of all classes into a unified set, where each element is a tuple of (class, bounding box). This means that multiple class predictions can be included for the

[1] `https://github.com/chemprop/chemprop`

[2] We use tf_efficientdet_d2 from `https://github.com/rwightman/efficientdet-pytorch`.

same bounding box (i.e., there is class uncertainty), and multiple bounding boxes can be found for the same class (i.e., there are multiple objects in one image). We define true positives as predictions that have an intersection over union (IoU) value > 0.5 with a gold bounding box annotation, *and* that match the annotation's class.

Entity extraction. Entity extraction is a popular NLP task. Given a sentence, such as "*Barack Obama was born in Hawaii*," the goal is to identify and classify all named entities that appear—i.e., ("Barack Obama", Person) and ("Hawaii", Location). We use the CoNLL NER dataset (Tjong Kim Sang and De Meulder, 2003), and extract $1k$ examples for calibration out of the $3.3k$ development set, and report test results on the $3.5k$ test set. For our base model, we use the entity extraction module of PURE (Zhong and Chen, 2021), that predicts span scores with a classifier head on top of Albert-base (Lan et al., 2020) contextual embeddings. The classification head has two non-linear layers and uses the learned contextual embeddings of the span start and end tokens, concatenated with a learned span width embedding. We train the model on the training set of the CoNLL NER dataset. We use the official code repository[3] and the following parameters: $1e{-}5$ learning rate, $5e{-}4$ task learning rate, 32 batch size, and 100 context window. Similar to our object detection task, we treat exact span predictions of the correct category as true positives, and any other entity predictions as false positives. As illustrated in Table 4.1, a fairly large number of sentences do not contain any entities at all, while other sentences may contain several.

D.3.1 Definition of size-stratified false positive violation

The size-stratified false positive (SSFP) violation measures the worst-case violation of our metric of interest (i.e., expectation or probability), conditioned on the *size* of the output set $\mathcal{C}$. Specifically, SSFP_k and $\mathrm{SSFP}_{k,\delta}$ are defined as follows:

[3] `https://github.com/princeton-nlp/PURE`.

$$\begin{aligned}
&\mathrm{SSFP}_k(\mathcal{C}, \{\mathcal{A}\}_{s=1}^a) := && \text{(D.32)}\\
&\quad \sup_s \max\left(\widehat{\mathbb{E}}\Big[\mathrm{FP}(Z_{n+1}, \mathcal{C}_\epsilon(X_{n+1})) \;\Big|\; \{|\mathcal{C}_\epsilon(X_{n+1})| \in \mathcal{A}_s\}\Big] - k, 0\right) \quad \text{and}\\
&\mathrm{SSFP}_{k,\delta}(\mathcal{C}, \{\mathcal{A}\}_{s=1}^a) := && \text{(D.33)}\\
&\quad \sup_s \max\left(\widehat{\mathbb{E}}\Big[\mathbf{1}\Big\{\mathrm{FP}(Z_{n+1}, \mathcal{C}_\epsilon(X_{n+1})) > k\Big\} \;\Big|\; \{|\mathcal{C}_\epsilon(X_{n+1})| \in \mathcal{A}_s\}\Big] - \delta, 0\right),
\end{aligned}$$

where $\{\mathcal{A}\}_{s=1}^a$ forms a partition of $\{1, \ldots, |\mathcal{Y}|\}$, and $\widehat{\mathbb{E}}$ denotes the empirical average over our test samples.

Following Angelopoulos et al. (2021b), we show that if conditional validity holds for our objectives, then validity also holds after stratifying by set-size. Poor SSFP is therefore a symptom of poor conditional validity.

In the following, we drop dependence on $n+1$ for clarity.

Proposition D.3.1 (Expectation case)**.** *Suppose* $\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_\epsilon(X)) \mid X = x] \leq k$ *for each* $x \in \mathcal{X}$*. Then for any* $\mathcal{A} \subset \{0, 1, 2, \ldots\}$,

$$\mathbb{E}\Big[\mathrm{FP}(Z, \mathcal{C}_\epsilon(X)) \mid \{|\mathcal{C}_\epsilon(X)| \in \mathcal{A}\}\Big] \leq k. \tag{D.34}$$

Proof.

$$\begin{aligned}
&\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_\epsilon(X)) \mid \{|\mathcal{C}_\epsilon(X)| \in \mathcal{A}\}]\\
&\quad = \frac{\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_\epsilon(X)) \cdot \mathbf{1}\{|\mathcal{C}_\epsilon(X)| \in \mathcal{A}\}]}{\mathbb{P}(|\mathcal{C}_\epsilon(X)| \in \mathcal{A})}\\
&\quad = \frac{\mathbb{E}[\mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_\epsilon(X)) \cdot \mathbf{1}\{|\mathcal{C}_\epsilon(X)| \in \mathcal{A}\} \mid X = x]]}{\mathbb{P}(|\mathcal{C}_\epsilon(X)| \in \mathcal{A})}\\
&\quad = \frac{\int_x \mathbb{E}[\mathrm{FP}(Z, \mathcal{C}_\epsilon(X)) \mid X = x] \cdot \mathbf{1}\{|\mathcal{C}_\epsilon(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_\epsilon(X)| \in \mathcal{A})}\\
&\quad \leq \frac{\int_x k \cdot \mathbf{1}\{|\mathcal{C}_\epsilon(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_\epsilon(X)| \in \mathcal{A})}\\
&\quad = k.
\end{aligned} \tag{D.35}$$

□

Proposition D.3.2 (Probability case.). *Suppose* $\mathbb{P}(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid X = x) \geq 1-\delta$ *for each* $x \in \mathcal{X}$*. Then for any* $\mathcal{A} \subset \{0, 1, 2, \ldots\}$*,*

$$\mathbb{P}\Big(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid \{|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A}\}\Big) \geq 1 - \delta. \tag{D.36}$$

Proof.

$$\begin{aligned} &\mathbb{P}(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid \{|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A}\}) \\ &\qquad = \frac{\int_x \mathbb{P}(\mathrm{FP}(Z, \mathcal{C}_{k,\delta}(X)) \leq k \mid X = x) \cdot \mathbf{1}\{|\mathcal{C}_{k,\delta}(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A})} \\ &\qquad \geq \frac{\int_x (1-\delta) \cdot \mathbf{1}\{|\mathcal{C}_{k,\delta}(x)| \in \mathcal{A}\} d\mathbb{P}(x)}{\mathbb{P}(|\mathcal{C}_{k,\delta}(X)| \in \mathcal{A})} \\ &\qquad = 1 - \delta. \end{aligned} \tag{D.37}$$

□

D.4 Additional experimental details

In this section we provide additional discussion of our experimental results.

D.4.1 Non-smoothness of SSFP results

The top rows of Figure 4-3 and Figure 4-2 plot the *worst-case* size-stratified violation, that is the worst-case exceedance of k, conditioned on the prediction set being a particular size. As can be seen from the plots, this can "spike" at different values of ϵ. We provide an interpretation here. As the false positive allowance k grows, the calibration threshold rises, but not all prediction sets necessarily grow in size at the same rate. Therefore, the "worst" sets experience discrete jumps in total false positives at various increments of k. This is most severe when the number of true positives is naturally very low, as in entity extraction, so that increases in set size often lead to increases in false positives (in the worst case). The exact behavior depends on the score function used. In the meantime, between these jumps, the worst case deviation will decrease, as deviation is measured as max($\mathbb{E}$[false positives | set size] −

$k, 0)$, where k is linearly increasing. The *average* deviation is much smoother, but less interesting for comparing various methods, which is why we focus on SSFP.

D.5 Other practical considerations and limitations

In this section we address a number of practical considerations, limitations, and extensions for our FP-CP method.

D.5.1 Choosing a suitable k

An outstanding question a practitioner faces is how to choose the value of k for k-FP and (k, δ)-FP objectives. The value of k in our method has a reliable and easy interpretation: it is the total number of incorrect answers. For many tasks, such as in-silico screening, there is a direct relation between the number of noisy predictions (e.g., failed experiments conducted during wet-lab validation) and total "wasted" cost. Therefore, for example, given some approximate budget Q and cost per noisy prediction c, a reasonable approach is to then set $k \approx Q/c$. Of course, the advantage of our approach is that the user may set k to whatever they wish—this might change based on their needs, and is not part of our algorithm.

D.5.2 Choosing between k-fwer and fdr control

A related question to D.5.1 is when to target k-FWER (i.e., our k-FP and (k, δ)-FP objectives) or FDR (e.g., using Angelopoulos et al. (2021a)). This choice is well discussed in the multiple testing literature (Lehmann and Romano, 2005a; Romano and Wolf, 2007; Gold et al., 2009). An important aspect to consider is the size of the label space $\mathcal{Y}$, natural rate of true and false positives, and the efficiency of the base model at separating true positives from false positives. When the total number of true positives is large and $|\mathcal{Y}|$ is large then it is reasonable to control the FDR. If, however, the natural rate of true positives is low, or they are not well separated from false positives, then the FDR can be high and hard to control. This is especially true for smaller prediction sets (as the ratio of positive to negative labels can be quickly driven down even with the addition of only a few false positives). For illustration, suppose for a given example there is one true positive that is ranked 10th by the base model. For many applications, 10 total predictions (with 9 false positives) is acceptable. Yet, the lowest FDR cutoff that allows for this positive to be discoverable is 0.9 (which, for other examples, may allow for hundreds of false positives—an outcome which is undesirable for some applications, even given a high number of accompanying true positives). To satisfy a lower FDR, the algorithm must output an empty set (with FDR = 0). This remains true even if there are a few (but not many) other true positives: for instance, in the previous example, if predictions 10-20 were also all true positives then the lowest FDR is still only 0.5—specifying a FDR tolerances any lower than this would force an empty set prediction.

D.5.3 Learning more expressive set functions

Our choice of DeepSets model is motivated by its property of being a universal approximator for continuous set functions, and by its efficient architecture. Of course, its realized accuracy depends on its exact parameterization and optimization. In terms of input features, in §4.3.2, we chose a simplistic $\phi(x, y_c)$ for two reasons: (1) we view it's low complexity as an advantage (practitioners can simply plug-in individual multi-label probabilities, or other scalar conformity scores, that most out-of-the-box methods provide into a general framework without having to do any more work for providing additional features), and (2) it is easy to train this light-weight model on smaller amounts of data. Still, this approach can discard potentially helpful information about the input x, and any dependencies between labels y_c and y_c'. For example, if y_c and y_c' are mutually exclusive, then the number of false positives if both are included in $\mathcal{S}$ is at least 1. Using more expressive ϕ that is able to capture and take advantage of this sort of side information about x and y_c is a subject for future work.

D.5.4 Constructing non-nested candidate sets

We choose to construct nested prediction sets because they are efficient and effective. It is useful to emphasize, however, that nestedness is not necessary for our calibration framework: our procedure still works even when candidate sets are not nested. It only relies on $\mathrm{FP}_{\max}$ remaining monotonic in t, which is preserved even for non-nested candidate sets. That said, generally speaking, considering individual candidates in the order of individual likelihood is a good strategy: this maximizes the expected number of true positives in a set of fixed size. Of course, we are not ranking by the true marginal likelihood, but rather the estimate, $p_\theta(y_c \in Z \mid x)$, and this may introduce some error. In theory, the set function $\mathcal{F}$ may be able to identify higher quality outputs sets $\mathcal{S} \in 2^{\mathcal{Y}}$ by jointly considering all of the included elements (rather than ranking them one-by-one). That said, an unconstrained search process over $2^{\mathcal{Y}}$ is expensive. Furthermore, identifying the final output set with maximal TPR, as we show we do in Proposition 4.3.8, is no longer trivial. Nevertheless, this is a promising area for future work, can potentially be combined with efficient search or pruning methods (e.g., such as in Fisch et al. (2021a), see also Chapter 3).

E

Appendix for Chapter 5

E.1 Proofs

Symbol	Meaning
k	The number of in-task examples used for few-shot learning.
ϵ	The stipulated performance tolerance.
δ	The stipulated secondary confidence tolerance for calibration conditional validity.
t	The total number of auxiliary tasks.
$\mathcal{T}$	The space of potential tasks to be solved in a few-shot learning setting.
$\mathcal{X} \times \mathcal{Y}$	The joint input (X's) and output (Y's) space.
$\mathcal{I}_{\text{train}}$	The set of auxiliary tasks used for meta-learning nonconformity scores and quantile predictors.
$\mathcal{I}_{\text{cal}}$	The set of auxiliary tasks used to calibrate the quantile predictor.
T_{t+1}	The target few-shot test task to be solved.
$\mathcal{N}$	A meta-learned nonconformity measure.
$\mathcal{P}_{1-\epsilon}$	A meta-learned regressor of the $1-\epsilon$ quantile of $\mathcal{N}$'s scores on T_{t+1} given k in-task samples.
$\widehat{V}_{i,j}^{(x,y)}$	The meta nonconformity score for example j of task i, given the current candidate output (x, y).
$F_i, \widehat{F}_{m_i}$	The true vs. m_i-sample empirical distribution function over nonconformity scores of task i.
$\widehat{Q}_i$, Quantile$(1-\epsilon, F_i)$	The predicted vs. true nonconformity score $1-\epsilon$ quantiles for task i.
$\widehat{\Lambda}(1-\epsilon, \mathcal{I}_{\text{cal}})$	The $1-\epsilon$ meta quantile correction factor, computed using calibration tasks.
$\mathcal{C}_\epsilon$, $\mathcal{M}_\epsilon$	Output label sets for standard and meta conformal prediction, respectively, at level $1-\epsilon$.

Table E.1.1: Definitions of common notations used in this chapter.

E.1.1 Proof of Lemma 5.3.2

Proof. Let the event $\{T_i = t_i\}$ indicate that task i has a quantile prediction $\{\widehat{Q}_i = \hat{q}_i)$ and CDF $\{F_i = f_i\}$ over meta nonconformity scores. For convenience, assume tasks T_i, $i \in \mathcal{I}_{\text{cal}}$ and T_{t+1} are indexed contiguously as $i = 1, \ldots, n+1$. Let

$$\begin{aligned} \widehat{\Lambda}'(\beta) &:= \inf\left\{\lambda \colon \frac{1}{n+1}\sum_{i=1}^{n+1} F_i\Big(\widehat{Q}_i + \lambda\Big) \geq \beta\right\}, \quad \text{and} \\ \widehat{\Lambda}(\beta) &:= \inf\left\{\lambda \colon \frac{1}{n+1}\sum_{i=1}^{n} F_i\Big(\widehat{Q}_i + \lambda\Big) \geq \beta\right\}. \end{aligned} \tag{E.1}$$

where $\widehat{\Lambda}(\beta)$ is the same as $\widehat{\Lambda}(\beta, \mathcal{I}_{\text{cal}})$ as per Eq. (5.8). As CDFs, F_i are non-decreasing, bounded in $[0, 1]$, and right-continuous. Therefore, both $\widehat{\Lambda}(\beta)$ and $\widehat{\Lambda}'(\beta)$ are well-defined. Furthermore, since $F_i \geq 0$, we have that $\widehat{\Lambda}'(\beta) \leq \widehat{\Lambda}(\beta)$, since

$$\frac{1}{n+1}\sum_{i=1}^{n} F_i(\widehat{Q}_i + \lambda) \geq \beta \Longrightarrow \frac{1}{n+1}\sum_{i=1}^{n+1} F_i(\widehat{Q}_i + \lambda) \geq \beta. \tag{E.2}$$

Next, denote by E_t the event that $\{T_1, \ldots, T_{n+1}\} = \{t_1, \ldots, t_{n+1}\}$, i.e., we observe an unordered set of task values. Exchangeability of tasks T_i implies that

$$\mathbb{P}(T_{n+1} = t_i \mid E_t) = \frac{1}{n+1}, \tag{E.3}$$

and, accordingly, that the distribution of $T_{n+1} \mid E_t$ is uniform on the set $\{t_1, \ldots, t_{n+1}\}$. For convenience, let $V_i := V_{i,k+1}^{\text{test}}$. For any constant $\lambda \in \mathbb{R}$,

$$\begin{aligned} \mathbb{P}(V_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid E_t) &= \sum_{i=1}^{n+1} \mathbb{P}(V_{n+1} \leq \widehat{Q}_{n+1} + \lambda, T_{n+1} = t_i \mid E_t) \\ &= \sum_{i=1}^{n+1} \mathbb{P}(V_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid T_{n+1} = t_i)\mathbb{P}(T_{n+1} = t_i \mid E_t) \\ &= \frac{1}{n+1}\sum_{i=1}^{n+1} \mathbb{P}(V_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid T_{n+1} = t_i). \end{aligned} \tag{E.4}$$

From our definition of the event $\{T_{n+1} = t_i\} = \{\widehat{Q}_{n+1} = \hat{q}_i, F_{n+1} = f_i\}$, we can rewrite

$$\mathbb{P}(V_{n+1} \leq \widehat{Q}_{n+1} + \lambda \mid E_t) = \frac{1}{n+1} \sum_{i=1}^{n+1} f_i(\hat{q}_i + \lambda). \tag{E.5}$$

Conditioned on E_t, $\widehat{\Lambda}'(\beta)$ is a constant. Right-continuity of f_i then implies that

$$\frac{1}{n+1} \sum_{i=1}^{n+1} f_i(\hat{q}_i + \widehat{\Lambda}'(\beta) \mid E_t) \geq \beta. \tag{E.6}$$

Since $\widehat{\Lambda}'(\beta) \mid E_t \leq \widehat{\Lambda}(\beta) \mid E_t$ and F_i are non-decreasing,

$$\frac{1}{n+1} \sum_{i=1}^{n+1} f_i(\hat{q}_i + \widehat{\Lambda}(\beta) \mid E_t) \geq \beta, \tag{E.7}$$

Finally, because this is true for any E_t, we can marginalize

$$\begin{aligned}
\mathbb{P}(V_{n+1} \leq \widehat{Q}_{n+1} + \widehat{\Lambda}(\beta)) &= \int_{E_t} \mathbb{P}(V_{n+1} \leq \widehat{Q}_{n+1} + \widehat{\Lambda}(\beta) \mid E_t)\; d\mathbb{P}(E_t) \\
&= \int_{E_t} \frac{1}{n+1} \sum_{i=1}^{n+1} f_i(\hat{q}_i + \widehat{\Lambda}(\beta) \mid E_t) d\mathbb{P}(E_t) \\
&\geq \beta \int_{E_t} d\mathbb{P}(E_t) \\
&= \beta.
\end{aligned} \tag{E.8}$$

N.B. See Chapter 6 for related analysis of more general monotonic risks, which follows the same proof strategy. □

E.1.2 Proof of Theorem 5.3.3

Proof. Let $V_i := V_{i,k+1}^{\text{test}}$. By definition, $Y_{t+1}^{\text{test}} \in \mathcal{M}_\epsilon(X_{t+1}^{\text{test}}) \iff V_{t+1} \leq \widehat{Q}_{t+1} + \widehat{\Lambda}(1 - \epsilon; \mathcal{I}_{\text{cal}})$. As $\mathcal{N}$ and $\mathcal{P}$ are trained on the disjoint proper training set $\mathcal{I}_{\text{train}}$, they preserve exchangeability, and produce exchangeable $\widehat{Q}_i$. We then apply Lemma 5.3.2. □

E.1.3 Proof of Proposition 5.3.5

Intuitively, as the predicted quantiles become more accurate, they require less correction during calibration. On the target task T_{t+1}, *under-coverage* happens when the predicted quantile/threshold, $\widehat{Q}_{t+1} + \widehat{\Lambda}(1-\epsilon, \mathcal{I}_{\text{cal}})$ is too low. We will show that as $k \to \infty$, this happens with probability 0.

For convenience, let $V_i := V^{\text{test}}_{i,k+1}$. Let tasks T_i, $i \in \mathcal{I}_{\text{cal}}$ and T_{t+1} be indexed contiguously as $i = 1, \ldots, n+1$ Let $\widehat{Q}^{(k)}_i$ refer to $\mathcal{P}_\beta(Z_{i,1:k}; \phi_{\text{meta}})$, the predicted quantile for task i using k in-task samples. We will also drop conditioning on the target task $T_{n+1} = t_{n+1}$ with the understanding that it applies on all probabilities.

Lemma E.1.1. *Under the same conditions as Proposition 5.3.5,*

$$\lim_{k\to\infty} \mathbb{P}\left(F_{n+1}\left(\widehat{Q}^{(k)}_{n+1} + \widehat{\Lambda}(\beta, \mathcal{I}_{\text{cal}})\right) < \beta\right) = 0. \tag{E.9}$$

Proof. By assumption, F_i are continuous and $\widehat{Q}^{(k)}_i \xrightarrow{p} \text{Quantile}(\beta, F_i)$. The continuous mapping theorem then gives

$$F_{n+1}\Big(\widehat{Q}^{(k)}_{n+1} + \widehat{\Lambda}(\beta, \mathcal{I}_{\text{cal}})\Big) \xrightarrow{p} F_{n+1}\Big(\text{Quantile}(\beta, F_{n+1}) + \Lambda(\beta, \mathcal{I}_{\text{cal}})\Big), \tag{E.10}$$

where

$$\Lambda(\beta, \mathcal{I}_{\text{cal}}) := \inf\left\{\lambda \colon \frac{1}{n+1}\sum_{i=1}^{n} F_i\Big(\text{Quantile}(\beta, F_i) + \lambda\Big) \geq \beta\right\}. \tag{E.11}$$

As this implies convergence in distribution,

$$\begin{aligned}
&\lim_{k\to\infty} \mathbb{P}\Big(F_{n+1}\Big(\widehat{Q}^{(k)}_{n+1} + \widehat{\Lambda}(\beta, \mathcal{I}_{\text{cal}})\Big) \geq \beta\Big) \\
&\qquad = \mathbb{P}\Big(F_{n+1}\Big(\text{Quantile}(\beta, F_{n+1}) + \Lambda(\beta, \mathcal{I}_{\text{cal}})\Big) \geq \beta\Big) \\
&\qquad \overset{(i)}{\geq} \mathbb{P}\Big(F_{n+1}\Big(\text{Quantile}(\beta, F_{n+1})\Big) \geq \beta \mid \Lambda(\beta, \mathcal{I}_{\text{cal}}) \geq 0\Big)\mathbb{P}\Big(\Lambda(\beta, \mathcal{I}_{\text{cal}}) \geq 0\Big) \\
&\qquad \overset{(ii)}{=} \mathbb{P}\Big(F_{n+1}\Big(\text{Quantile}(\beta, F_{n+1})\Big) \geq \beta\Big)\mathbb{P}\Big(\Lambda(\beta, \mathcal{I}_{\text{cal}}) \geq 0\Big) \\
&\qquad \overset{(iii)}{=} \mathbb{P}\Big(\Lambda(\beta, \mathcal{I}_{\text{cal}}) \geq 0\Big),
\end{aligned} \tag{E.12}$$

where (i) is from F_{n+1} being non-decreasing, (ii) is from the independence of tasks $\{T_i\}_{i=1}^{n+1}$, and (iii) is from $F_{n+1}(\text{Quantile}(\beta, F_{n+1})) = \beta$, for continuous F_{n+1}.

Finally, since F_i are monotonic increasing and continuous,

$$\begin{aligned}
\mathbb{P}\Big(\Lambda(\beta, \mathcal{I}_{\text{cal}}) \geq 0\Big) &= 1 - \mathbb{P}\Big(\exists \lambda < 0 \colon \frac{1}{n+1} \sum_{i=1}^{n} F_i\Big(\text{Quantile}(\beta, F_i) + \lambda) \geq \beta\Big) \\
&\geq 1 - \mathbb{P}\Big(\frac{1}{n+1} \sum_{i=1}^{n} F_i\Big(\text{Quantile}(\beta, F_i)) \geq \beta\Big) \\
&= 1 - \mathbb{P}\Big(\frac{n}{n+1} \beta \geq \beta\Big) \\
&= 1.
\end{aligned} \tag{E.13}$$

□

We now proceed to prove Proposition 5.3.5.

Proof. Y_{t+1}^{test} is included in $\mathcal{M}_\epsilon(X_{t+1}^{\text{test}})$ iff $V_{t+1} \leq \widehat{Q}_{t+1}^{(k)} + \Lambda(1-\epsilon, \mathcal{I}_{\text{cal}})$, thus

$$\begin{aligned}
\mathbb{P}\Big(Y_{t+1}^{\text{test}} \notin \mathcal{M}_\epsilon(X_{t+1}^{\text{test}})\Big) &= 1 - \mathbb{P}\Big(Y_{t+1}^{\text{test}} \in \mathcal{M}_\epsilon(X_{t+1}^{\text{test}})\Big) \\
&= 1 - \mathbb{P}(V_{t+1} \leq \widehat{Q}_{t+1}^{(k)} + \Lambda(1-\epsilon, \mathcal{I}_{\text{cal}})) \\
&= 1 - F_{t+1}(\widehat{Q}_{t+1}^{(k)} + \Lambda(1-\epsilon, \mathcal{I}_{\text{cal}})).
\end{aligned} \tag{E.14}$$

Applying Lemma E.1.1 gives

$$\begin{aligned}
&\lim_{k \to \infty} \mathbb{P}\Big(\mathbb{P}\Big(Y_{t+1}^{\text{test}} \notin \mathcal{M}_\epsilon(X_{t+1}^{\text{test}})\Big) \leq \epsilon\Big) \\
&\qquad = \lim_{k \to \infty} \mathbb{P}\Big(F_{t+1}(\widehat{Q}_{t+1}^{(k)} + \Lambda(1-\epsilon, \mathcal{I}_{\text{cal}})) \geq 1 - \epsilon\Big) \\
&\qquad = 1.
\end{aligned} \tag{E.15}$$

□

E.1.4 Proof of Proposition 5.3.7

Proof. Let $\widehat{F}_{m_i}$ be the m_i-sample ECDF for T_i. Define the empirical correction, $\widehat{\Lambda}'(\beta, \mathcal{I}_{\text{cal}})$, when plugging in $\widehat{F}_{m_i}$ as

$$\widehat{\Lambda}^{\text{emp}}(\beta, \mathcal{I}_{\text{cal}}) := \inf \left\{ \lambda \colon \frac{1}{|\mathcal{I}_{\text{cal}}| + 1} \sum_{i \in \mathcal{I}_{\text{cal}}} \widehat{F}_{m_i}\Big(\widehat{Q}_i + \lambda\Big) \geq \beta \right\} \tag{E.16}$$

where the ECDF is calculated as

$$\widehat{F}_{m_i} := \sum_{j=1}^{m_i} \mathbf{1}\{V_{i,k+j}^{\text{test}} \leq \widehat{Q}_i + \lambda\},$$

where $V_{i,k+j}^{\text{test}}$ are i.i.d.

We proceed in two parts. First, we prove that if, for some choice of τ, the error incurred by using the ECDF instead of the true CDF is bounded by τ w.p. $1 - \delta$, then $\mathcal{M}_{\epsilon-\tau}$ is (δ, ϵ) valid. Due to monotonicity of coverage in the correction factor, it is sufficient to prove that $\widehat{\Lambda}^{\text{emp}}(\beta + \tau, \mathcal{I}_{\text{cal}}) \geq \widehat{\Lambda}(\beta + \tau, \mathcal{I}_{\text{cal}})$ w.p. $1 - \delta$. Second, we show that this holds for τ defined per Eq. (5.14).

(1) For ease of notation, let

$$B(\lambda) := \frac{1}{|\mathcal{I}_{\text{cal}}| + 1} \sum_{i \in \mathcal{I}_{\text{cal}}} \widehat{F}_{m_i}(\widehat{Q}_i + \lambda), \text{ and} \tag{E.17}$$

$$\widehat{B}(\lambda) := \frac{1}{|\mathcal{I}_{\text{cal}}| + 1} \sum_{i \in \mathcal{I}_{\text{cal}}} F_i(\widehat{Q}_i + \lambda). \tag{E.18}$$

Both $B(\lambda)$ and $\widehat{B}(\lambda)$ are non-decreasing in λ.

Next, assume that (to be proved) that for some $\tau > 0$

$$\mathbb{P}(\sup_{\lambda} \widehat{B}(\lambda) - B(\lambda) \leq \tau) \geq 1 - \delta. \tag{E.19}$$

Then w.p. $1 - \delta$,

$$\begin{aligned}\widehat{\Lambda}^{\text{emp}}(\beta + \tau, \mathcal{I}_{\text{cal}}) &= \inf \Big\{ \lambda \colon \widehat{B}(\lambda) \geq \beta + \tau \Big\} \\ &\overset{(i)}{\geq} \inf \Big\{ \lambda \colon B(\lambda) + \tau \geq \beta + \tau \Big\} \\ &= \inf \Big\{ \lambda \colon B(\lambda) \geq \beta \Big\} \\ &= \widehat{\Lambda}(\beta, \mathcal{I}_{\text{cal}})\end{aligned} \tag{E.20}$$

where (i) is due to our assumption that $\widehat{B}(\lambda) \leq B(\lambda) + \tau \ \forall \lambda$, and the fact that both $\widehat{B}(\lambda)$ and $B(\lambda)$ are non-decreasing in λ. Since $\widehat{\Lambda}^{\text{emp}}(\beta + \tau, \mathcal{I}_{\text{cal}}) \geq \widehat{\Lambda}(\beta + \tau, \mathcal{I}_{\text{cal}})$, it follows by monotonicity that coverage is preserved (see also the proof for Lemma 5.3.2).

(2) We now prove Eq. (E.19). Given an m-sample ECDF, $\widehat{F}_m(u)$, for some random variable U, the Dvoretsky-Kiefer-Wolfowitz inequality (Dvoretzky et al., 1956) allows us to build an interval for the value of the true distribution function, $F(u)$, where

$$\mathbb{P}\left(\sup_{u \in \mathbb{R}} |\widehat{F}_m(u) - F(u)| > \gamma \right) \leq 2e^{-2n\gamma^2}. \tag{E.21}$$

Alternatively stated, w.p. $\geq 1 - \alpha$, $F(u) \in [\widehat{F}_m(u) - \gamma, \widehat{F}_m(u) + \gamma]$, where $\gamma = \sqrt{\frac{\log \frac{2}{\alpha}}{2n}}$.

Let $Y_i := \widehat{F}_{m_i}(V_i \leq \widehat{Q}_i + \lambda) - F_i(V_i \leq \widehat{Q}_i + \lambda)$. For convenience tasks T_i, $i \in \mathcal{I}_{\text{cal}}$ and T_{t+1} are indexed contiguously as $i = 1, \ldots, n+1$. The difference, $A - B$, is then equivalent to $\frac{1}{n+1} \sum_{i=1}^{n} Y_i$, where Y_i are i.i.d., $\mathbb{E}[Y_i] = 0$, and Y_i is bounded with probability $1 - \alpha$. Applying Hoeffding's inequality gives

$$\begin{aligned}\mathbb{P}\Big(\frac{1}{n+1} \sum_{i=1}^{n} Y_i < \tau \Big) &\geq \mathbb{P}\Big(\sum_{i=1}^{n} Y_i < n\tau, \bigcap_{i=1}^{n} Y_i \in [-\gamma_i, \gamma_i] \Big) \\ &\geq \mathbb{P}\Big(\sum_{i=1}^{n} Y_i < n\tau \ \Big| \ \bigcap_{i=1}^{n} Y_i \in [-\gamma_i, \gamma_i] \Big) \mathbb{P}\Big(\bigcap_{i=1}^{n} Y_i \in [-\gamma_i, \gamma_i] \Big) \\ &\geq \left(1 - e^{-\frac{2n^2\tau^2}{\sum_{i=1}^{n}(2\gamma_i)^2}} \right) \left(1 - \alpha \right)^n.\end{aligned} \tag{E.22}$$

Solving for τ given the target $1-\delta$ error probability yields

$$\tau \geq \sqrt{\frac{-2}{n^2}\Big(\sum_{i=1}^{n}\gamma_i^2\Big)\log\Big(1-\frac{1-\delta}{(1-\alpha)^n}\Big)} \tag{E.23}$$

This holds for any choice of α. □

E.2 Meta conformal prediction details

E.2.1 Meta-Learning algorithms

Prototypical networks (Snell et al., 2017)

We use prototypical networks for our classification tasks. We assume that for each task we have N total classes with K examples per class (for a total of $k = N \times K$ training examples). In this model, an encoder, $\mathbf{h} = \mathrm{enc}(x;\theta)$ is trained to produce vector representations. Thereafter, a "prototype" for each class is computed by averaging the representations of all instances of that class. Let S_j denote the support set of training examples for class j. Then the prototype $\mathbf{c}_j$ is

$$\mathbf{c}_j := \frac{1}{|S_j|}\sum_{(x_i,y_i)\in S_j}\mathrm{enc}(x_i;\theta). \tag{E.24}$$

The likelihood of each class is then calculated using a softmax over the euclidean distance to each prototype:

$$p_\theta(y = j \mid x) := \frac{\exp(-d(\mathbf{c}_j, \mathrm{enc}(x;\theta)))}{\sum_{j'}\exp(-d(\mathbf{c}_{j'}, \mathrm{enc}(x;\theta)))}, \tag{E.25}$$

where $d(\cdot,\cdot)$ denotes the euclidean distance.

During training, random "episodes" are created by sampling N classes from the training set. For each class, K examples are randomly sampled to construct the prototypes. An additional Q examples are then sampled to simulate queries. The objective is to then minimize the cross entropy loss across queries.After training, we use $-p_\theta(y =$

$j \mid x)$ as defined in Eq. (E.25) as the nonconformity measure for label $y = j$.

Differentiable ridge regression (Bertinetto et al., 2019)

We use differentiable ridge regression for our regression tasks. We assume that for each task we have k labeled (x_i, y_i) pairs, where $y \in \mathbb{R}$. In this model, like the prototypical networks, an encoder, $\mathbf{h} = \text{enc}(x; \theta)$, is trained to produce vector representations of dimension d. We then solve a least-squares regression to obtain our prediction, $\hat{y} = \mathbf{w} \cdot \text{enc}(x; \theta)$, where

$$\mathbf{w} = X^\top (XX^\top + \lambda I)^{-1} Y \tag{E.26}$$

with $X \in \mathbb{R}^{k \times d}$, $Y \in \mathbb{R}^k$, and λ a meta regularization parameter that we optimize. We optimize MSE by back-propagating through the least-squares operator to the encoder. We train using the same episode-based procedure that we described for the prototypical networks. After training, we use the absolute error, $|\hat{y} - y|$, as the nonconformity score for candidate $y \in \mathbb{R}$.

Deep sets (Zaheer et al., 2017)

We use a simple deep sets architecture for all of our quantile predictors. Deep sets are of the form

$$f(X) := \text{dec}\Big(\sum_{x \in X} \text{enc}(x; \phi_1); \phi_2\Big) \tag{E.27}$$

where X is an input set of elements, enc is an element-wise encoder, and dec is a decoder that operates on the aggregated encoded set elements. Importantly, the deep sets model f is invariant to permutations of the elements in X.

E.2.2 Implementation details

Image classification. Each image is first resized to 84 × 84 pixels. We use a CNN encoder with 4 layers. Each layer contains a 3×3 convolution kernel with 64 channels and a padding of size 1, followed by batch normalization layer, ReLU activation, and a 2×2 max pooling filter. The final output is of size 1600, which we use to compute the prototypes and as the query representations. We train the model for 100 epochs with an Adam optimizer and a batch size of 256. In each epoch, we run 100 episodes in which we sample 10 support images and 15 query images per class.

Relation classification. We use GloVe (Pennington et al., 2014) word embeddings of size 50 to convert the sentence into vectors. To each word embedding, we also concatenate two learned position embeddings of size 5, where the positions are relative to the location of the two entities in the sentence. Thereafter, a 1D convolution is applied with 230 output channels, a kernel size of 3 and padding size 1, followed by a ReLU activation. Finally, a max pooling filter is applied. The resultant sentence representation of size 230 is used to compute the prototypes and query representations. We train the model for a total of $20k$ episodes with a SGD optimizer and a batch size of 32. In each episode, we sample 10 support sentences and 5 query sentences per class.

Chemical property prediction. Our ridge regression network uses directed message passing networks (Yang et al., 2019) to compute $\text{enc}(x; \theta)$. The message passing network uses graph convolutions to learn a deep molecular representation that is shared across property predictions. We also include additional RDKit features as inputs.[1] We map inputs with a FFNN with hidden size 200, and then apply 3 layers of graph convolutions with a hidden size of 256. Finally, we map the output representation to a hidden size of 16, and apply least-squares regression. We train the network using an Adam optimizer for 15 epochs with 8 meta episodes per batch, each with 32 queries (for a total batch size of 256).

Quantile prediction. For all of our quantile predictors, we use a 2-layer FFNN for both the element-wise encoder, $\text{enc}(\cdot; \phi_1)$, and the aggregated set decoder, $\text{dec}(\cdot; \phi_2)$. Each FFNN has a hidden size of 256 and uses ReLU activations. We train the network using an Adam optimizer for 15 epochs with batch size 64.

[1] www.rdkit.org

E.2.3 Training strategy

We adopt a cross-fold procedure for training our meta nonconformity measure $\mathcal{N}$ and meta quantile predictor $\mathcal{P}_{1-\epsilon}$ in a data efficient way, as outlined in §5.3.2. Figure 5-4 illustrates this cross-fold process, in which we train a meta-nonconformity measure on each training fold and aggregate their predictions as input data for the quantile predictor. Since we train in a cross-fold manner but ultimately use a meta-nonconformity measure $\mathcal{N}$ that is trained on *all* of the training data, there is a train-test mismatch in the data supplied to the quantile predictor. Nevertheless, any error induced by this discrepancy (and any other sources of error, for that matter) is handled during meta-calibration (§5.3.3). All experiments took 1-5 hours to run on an Nvidia 2080 Ti GPU. As absolute performance is not the primary goal of this work, little hyperparameter tuning was done (most hyperparameters were taken from prior work). Datasets are available for *mini*ImageNet[2], FewRel 1.0[3], and ChEMBL[4].

[2] https://github.com/yaoyao-liu/mini-imagenet-tools

[3] https://thunlp.github.io/1/fewrel1.html

[4] https://github.com/chemprop/chemprop

F

Appendix for Chapter 6

F.1 Proofs

F.1.1 Proof of Theorem 6.3.2

We use the following lemma on the approximate continuity of the empirical risk.

Lemma F.1.1 (Jump Lemma)**.** *In the setting of Theorem 6.3.2, any jumps in the empirical risk are bounded, i.e.,*

$$\sup_{\lambda} J\Big(\widehat{R}_n, \lambda\Big) \overset{a.s.}{\leq} \frac{B}{n}. \tag{F.1}$$

Proof. By boundedness, the maximum contribution of any point to the jump is $\frac{B}{n}$, so

$$\exists \lambda : \; J\Big(\widehat{R}_n, \lambda\Big) > \frac{B}{n} \Longrightarrow \exists \lambda : \; J(L_i, \lambda) > 0 \text{ and } J(L_j, \lambda) > 0 \text{ for some } i \neq j. \tag{F.2}$$

Call $\mathcal{D}_i = \{\lambda : J(L_i, \lambda) > 0\}$ the sets of discontinuities in L_i. Since L_i is bounded monotone, $\mathcal{D}_i$ has countably many points. The union bound then implies that

$$\mathbb{P}\left(\exists \lambda : \; J(\widehat{R}_n, \lambda) > \frac{B}{n}\right) \leq \sum_{i \neq j} \mathbb{P}(\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset) \tag{F.3}$$

Rewriting each term of the right-hand side using tower property and law of total probability gives

$$\mathbb{P}\left(\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset\right) = \mathbb{E}\left[\mathbb{P}\Big(\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset \;\Big|\; \mathcal{D}_j\Big)\right] \tag{F.4}$$

$$\leq \mathbb{E}\left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}\left(\lambda \in \mathcal{D}_i \;\middle|\; \mathcal{D}_j\right)\right] = \mathbb{E}\left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}\left(\lambda \in \mathcal{D}_i\right)\right], \tag{F.5}$$

Where the second inequality is because the union of the events $\lambda \in \mathcal{D}_j$ is the entire sample space, but they are not disjoint, and the third equality is due to the independence between $\mathcal{D}_i$ and $\mathcal{D}_j$. Rewriting in terms of the jump function and applying the assumption $\mathbb{P}\left(J(L_i, \lambda) > 0\right) = 0$,

$$\mathbb{E}\left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}\left(\lambda \in \mathcal{D}_i\right)\right] = \mathbb{E}\left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}\left(J(L_i, \lambda) > 0\right)\right] = 0. \tag{F.6}$$

Chaining the above inequalities yields $\mathbb{P}\left(\exists\lambda : J(\widehat{R}_n, \lambda) > \frac{B}{n}\right) \leq 0$, so
$\mathbb{P}\left(\exists\lambda : J(\widehat{R}_n, \lambda) > \frac{B}{n}\right) = 0$. □

Proof of Theorem 6.3.2. If $L_i(\lambda_{\max}) \geq \alpha - 2B/(n+1)$, then $\mathbb{E}[L_{n+1}(\hat{\lambda})] \geq \alpha - 2B/(n+1)$. Throughout the rest of the proof, we assume that $L_i(\lambda_{\max}) < \alpha - 2B/(n+1)$. Define the quantity

$$\hat{\lambda}'' = \inf\left\{\lambda : \widehat{R}_{n+1}(\lambda) + \frac{B}{n+1} \leq \alpha\right\}. \tag{F.7}$$

Since $L_i(\lambda_{\max}) < \alpha - 2B/(n+1) < \alpha - B/(n+1)$, $\hat{\lambda}''$ exists almost surely. Deterministically, $\frac{n}{n+1}\widehat{R}_n(\lambda) \leq \widehat{R}_{n+1}(\lambda)$, which yields $\hat{\lambda} \leq \hat{\lambda}''$. Again since $L_i(\lambda)$ is non-increasing in λ,

$$\mathbb{E}\left[L_{n+1}\big(\hat{\lambda}''\big)\right] \leq \mathbb{E}\left[L_{n+1}\big(\hat{\lambda}\big)\right] \tag{F.8}$$

By exchangeability and the fact that $\hat{\lambda}''$ is a symmetric function of $L_1, \ldots, L_{n+1}$,

$$\mathbb{E}\left[L_{n+1}\big(\hat{\lambda}''\big)\right] = \mathbb{E}\left[\widehat{R}_{n+1}\big(\hat{\lambda}''\big)\right] \tag{F.9}$$

For the remainder of the proof we focus on lower-bounding $\widehat{R}_{n+1}\left(\hat{\lambda}''\right)$. We begin with the following identity:

$$\alpha = \widehat{R}_{n+1}\left(\hat{\lambda}''\right) + \frac{B}{n+1} - \left(\widehat{R}_{n+1}\left(\hat{\lambda}''\right) + \frac{B}{n+1} - \alpha\right). \tag{F.10}$$

Rearranging the identity,

$$\widehat{R}_{n+1}\left(\hat{\lambda}''\right) = \alpha - \frac{B}{n+1} + \left(\widehat{R}_{n+1}\left(\hat{\lambda}''\right) + \frac{B}{n+1} - \alpha\right). \tag{F.11}$$

Using the Jump Lemma to bound $\left(\widehat{R}_{n+1}\left(\hat{\lambda}''\right) + \frac{B}{n+1} - \alpha\right)$ below by $-\frac{B}{n+1}$ gives

$$\widehat{R}_{n+1}\left(\hat{\lambda}''\right) \geq \alpha - \frac{2B}{n+1}. \tag{F.12}$$

Finally, chaining together the above inequalities,

$$\mathbb{E}\left[L_{n+1}(\hat{\lambda})\right] \geq \mathbb{E}\left[\widehat{R}_{n+1}(\hat{\lambda}'')\right] \geq \alpha - \frac{2B}{n+1}. \tag{F.13}$$

□

F.1.2 Proof of Proposition 6.3.3

Proof. Without loss of generality, assume $B = 1$. Fix any $\epsilon' > 0$. Consider the following loss functions, which satisfy the conditions in Theorem 6.3.2:

$$L_i(\lambda) \overset{i.i.d.}{\sim} \begin{cases} 1 & \lambda \in [0, Z_i) \\ \frac{k}{k+1} & \lambda \in [Z_i, W_i) \\ 0 & \text{else} \end{cases}, \tag{F.14}$$

where $k \in \mathbb{N}$, the $Z_i \overset{i.i.d.}{\sim} \text{Uniform}(0, 0.5)$, the $W_i \overset{i.i.d.}{\sim} \text{Uniform}(0.5, 1)$ for $i \in \{1, ..., n + 1\}$ and $\alpha = \frac{k+1-\epsilon'}{n+1}$. Then, by the definition of $\hat{\lambda}$, we know

$$\widehat{R}_n\left(\hat{\lambda}\right) \leq \frac{k - \epsilon'}{n}. \tag{F.15}$$

If $n > k + 1$, $\widehat{R}(\lambda) \geq \frac{k}{k+1} > \frac{k}{n}$ whenever $\lambda \leq \frac{1}{2}$. Thus, we must have $\hat{\lambda} > \frac{1}{2}$. Since k is an integer and by (F.15), we know that $\left|\{i \in \{1, ..., n\} : L_i\left(\hat{\lambda}\right) > 0\}\right| \leq \lfloor (k + 1)(k - \epsilon')/k \rfloor \leq k$. This immediately implies that

$$\hat{\lambda} \geq W_{(n-k+1)}, \tag{F.16}$$

where $W_{(j)}$ denotes the j-th order statistic. Notice that for all $\lambda > \frac{1}{2}$,

$$R(\lambda) = \mathbb{E}\left[L_i(\lambda)\right] = \frac{k}{k+1}\mathbb{P}\left(W_i > \lambda\right) = \frac{k}{k+1} \cdot 2(1 - \lambda), \tag{F.17}$$

so $R\left(\hat{\lambda}\right) \leq \frac{k}{k+1} \cdot 2(1 - W_{(n-k+1)})$. Let $U_{(k)}$ be the k-th smallest order statistic of n i.i.d. uniform random variables on $(0, 1)$. Then, by symmetry and rescaling, $2(1 - W_{(n-k+1)}) \overset{d}{=} U_{(k)}$,

$$R\left(\hat{\lambda}\right) \preceq \frac{k}{k+1} U_{(k)},$$

where $\preceq$ denotes the stochastic dominance. It is well-known that $U_{(k)} \sim \text{Beta}(k, n + 1 - k)$ and hence

$$\mathbb{E}[R\left(\hat{\lambda}\right)] \leq \frac{k}{k+1} \cdot \frac{k}{n+1}.$$

Thus,

$$\alpha - \mathbb{E}\left[R\left(\hat{\lambda}\right)\right] \geq \frac{k + 1 - \epsilon}{n+1} - \frac{k^2}{(n+1)(k+1)} = \frac{1}{n+1} \cdot \frac{(2 - \epsilon')k + 1 - \epsilon'}{k+1}. \tag{F.18}$$

For any given $\epsilon > 0$, let $\epsilon' = \epsilon/2$ and $k = \lceil \frac{2}{\epsilon} - 1 \rceil$. Then

$$\frac{(2 - \epsilon')k + 1 - \epsilon'}{k+1} \geq 2 - \epsilon,$$

implying that

$$\alpha - \mathbb{E}\Big[R\big(\hat{\lambda}\big)\Big] \geq \frac{2-\epsilon}{n+1}.$$

□

F.1.3 Proof of Proposition 6.3.4

Proof of Proposition 6.3.4. Without loss of generality, we assume $B = 1$. Assume $\hat{\lambda}$ takes values in $[0,1]$ and $\alpha \in (1/(n+1), 1)$. Let $p \in (0,1)$, N be any positive integer, and $L_i(\lambda)$ be i.i.d. right-continuous piecewise constant (random) functions with

$$L_i(N/N) = 0, \quad (L_i(0/N), L_i(1/N), \ldots, L_i((N-1)/N)) \overset{i.i.d.}{\sim} \mathrm{Ber}(p). \tag{F.19}$$

By definition, $\hat{\lambda}$ is independent of L_{n+1}. Thus, for any $j = 0, 1, \ldots, N-1$,

$$\Big\{L_{n+1}(\hat{\lambda}) \mid \hat{\lambda} = j/N\Big\} \sim \mathrm{Ber}(p), \quad \Big\{L_{n+1}(\hat{\lambda}) \mid \hat{\lambda} = 1\Big\} \sim \delta_0. \tag{F.20}$$

Then,

$$\mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}\big)\Big] = p \cdot \mathbb{P}(\hat{\lambda} \neq 1) \tag{F.21}$$

Note that

$$\hat{\lambda} \neq 1 \Longleftrightarrow \min_{j \in \{0,\ldots,N-1\}} \frac{1}{n+1} \sum_{i=1}^{n} L_i(j/N) \leq \alpha - \frac{1}{n+1}. \tag{F.22}$$

Since $\alpha > 1/(n+1)$,

$$\begin{aligned}
\mathbb{P}(\hat{\lambda} \neq 1) = 1 - \mathbb{P}(\hat{\lambda} = 1) &= 1 - \mathbb{P}\left(\text{for all } j, \text{ we have } \frac{1}{n+1}\sum_{i=1}^{n} L_i(j/N) > \alpha - \frac{1}{n+1}\right) \\
&= 1 - \left(\sum_{k=\lceil (n+1)\alpha \rceil}^{n} \binom{n}{k} p^k (1-p)^{(n-k)}\right)^N \\
&= 1 - \Big(1 - \mathrm{BinoCDF}\big(n, p, \lceil (n+1)\alpha \rceil - 1\big)\Big)^N
\end{aligned}$$

As a result,

$$\mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}\big)\Big] = p\Bigg(1 - \Big(1 - \text{BinoCDF}\Big(n, p, \lceil (n+1)\alpha \rceil - 1\Big)\Big)^N\Bigg). \tag{F.23}$$

Now let N be sufficiently large such that

$$\Bigg(1 - \Big(1 - \text{BinoCDF}\Big(n, p, \lceil (n+1)\alpha \rceil - 1\Big)\Big)^N\Bigg) > p. \tag{F.24}$$

Then

$$\mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}\big)\Big] > p^2 \tag{F.25}$$

For any $\alpha > 0$, we can take p close enough to 1 to render the claim false. □

F.1.4 Proof of Theorem 6.3.6

Proof of Theorem 6.3.6. Define the *monotonized population risk* as

$$R^{\uparrow}(\lambda) = \sup_{t \geq \lambda} \mathbb{E}\Big[L_{n+1}(t)\Big] \tag{F.26}$$

Note that the independence of L_{n+1} and $\hat{\lambda}^{\uparrow}_n$ implies that for all n,

$$\mathbb{E}\Big[L_{n+1}\big(\hat{\lambda}^{\uparrow}_n\big)\Big] \leq \mathbb{E}\Big[R^{\uparrow}\big(\hat{\lambda}^{\uparrow}_n\big)\Big]. \tag{F.27}$$

Since $R^{\uparrow}$ is bounded, monotone, and one-dimensional, a generalization of the Glivenko-Cantelli Theorem given in Theorem 1 of (DeHardt, 1971) gives that uniformly over λ,

$$\lim_{n\to\infty} \sup_{\lambda} |\widehat{R}_n(\lambda) - R(\lambda)| \overset{a.s.}{\to} 0. \tag{F.28}$$

As a result,

$$\lim_{n\to\infty} \sup_{\lambda} |\widehat{R}^{\uparrow}_n(\lambda) - R^{\uparrow}(\lambda)| \overset{a.s.}{\to} 0, \tag{F.29}$$

which implies that

$$\lim_{n\to\infty} |\widehat{R}^{\uparrow}_n(\hat{\lambda}^{\uparrow}) - R^{\uparrow}(\hat{\lambda}^{\uparrow})| \overset{a.s.}{\to} 0. \tag{F.30}$$

By definition, $\widehat{R}^{\uparrow}(\hat{\lambda}^{\uparrow}) \leq \alpha$ almost surely and thus this directly implies

$$\limsup_{n\to\infty} R^{\uparrow}\left(\hat{\lambda}_n^{\uparrow}\right) \leq \alpha \quad \text{a.s..} \tag{F.31}$$

Finally, since for all n, $R^{\uparrow}\left(\hat{\lambda}_n^{\uparrow}\right) \leq B$, by Fatou's lemma,

$$\lim_{n\to\infty} \mathbb{E}\left[L_{n+1}\left(\hat{\lambda}_n^{\uparrow}\right)\right] \leq \limsup_{n\to\infty} \mathbb{E}\left[R^{\uparrow}\left(\hat{\lambda}_n^{\uparrow}\right)\right] \leq \mathbb{E}\left[\limsup_{n\to\infty} R^{\uparrow}\left(\hat{\lambda}_n^{\uparrow}\right)\right] \leq \alpha. \tag{F.32}$$

□

F.1.5 Proof of Proposition 6.5.1

Proof of Proposition 6.5.1. Let

$$\hat{\lambda}' = \inf\left\{\lambda : \frac{\sum_{i=1}^{n+1} w(X_i)L_i(\lambda)}{\sum_{i=1}^{n+1} w(X_i)} \leq \alpha\right\}. \tag{F.33}$$

Since $\inf_\lambda L_i(\lambda) \leq \alpha$, $\hat{\lambda}'$ exists almost surely. Using the same argument as in the proof of Theorem 6.3.1, we can show that $\hat{\lambda}' \leq \hat{\lambda}(X_{n+1})$. Since $L_{n+1}(\lambda)$ is non-increasing in λ,

$$\mathbb{E}[L_{n+1}(\hat{\lambda}(X_{n+1}))] \leq \mathbb{E}[L_{n+1}(\hat{\lambda}')]. \tag{F.34}$$

Let E be the multiset of loss functions $\{(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})\}$. Then $\hat{\lambda}'$ is a function of E, or, equivalently, $\hat{\lambda}'$ is a constant conditional on E. Lemma 3 of Tibshirani et al. (2019) implies that

$$(X_{n+1}, Y_{n+1}) \mid E \sim \sum_{i=1}^{n+1} \frac{w(X_i)}{\sum_{j=1}^{n+1} w(X_j)} \delta_{(X_j, Y_j)} \Longrightarrow L_{n+1} \mid E \sim \sum_{i=1}^{n+1} \frac{w(X_i)}{\sum_{j=1}^{n+1} w(X_j)} \delta_{L_i}$$

where δ_z denotes the Dirac measure at z. Together with the right-continuity of L_i, the above result implies

$$\mathbb{E}\left[L_{n+1}(\hat{\lambda}') \mid E\right] = \frac{\sum_{i=1}^{n+1} w(X_i)L_i(\hat{\lambda}')}{\sum_{i=1}^{n+1} w(X_i)} \leq \alpha. \tag{F.35}$$

The proof is then completed by the law of total expectation. □

F.1.6 Proof of Proposition 6.5.2

Proof. Define the vector $Z' = (Z_1', \dots, Z_n', Z_{n+1})$, where $Z_i' \overset{i.i.d.}{\sim} \mathcal{L}(Z_{n+1})$ for all $i \in [n]$. Let

$$\epsilon = \sum_{i=1}^{n} \mathrm{TV}(Z_i, Z_i'). \tag{F.36}$$

By sublinearity,

$$\mathrm{TV}(Z, Z') \leq \epsilon. \tag{F.37}$$

It is a standard fact that (F.37) implies

$$\sup_{f \in \mathcal{F}_{\mathbf{1}}} |\mathbb{E}[f(Z)] - \mathbb{E}[f(Z')]| \leq \epsilon, \tag{F.38}$$

where $\mathcal{F}_{\mathbf{1}} = \{f : \mathcal{Z} \mapsto [0,1]\}$. Let $\ell : \mathcal{Z} \times \Lambda \to [0, B]$ be a bounded loss function. Furthermore, let $g(z) = \ell(z_{n+1}; \hat{\lambda}(z_1, \dots, z_n))$. Since $g(Z) \in [0, B]$,

$$|\mathbb{E}[g(Z)] - \mathbb{E}[g(Z')]| \leq B\epsilon. \tag{F.39}$$

Furthermore, since $Z_1', \dots, Z_{n+1}'$ are exchangeable, we can apply Theorems 6.3.1 and 6.3.2 to $\mathbb{E}[g(Z')]$, recovering

$$\alpha - \frac{2B}{n+1} \leq \mathbb{E}[g(Z')] \leq \alpha. \tag{F.40}$$

A final step of triangle inequality implies the result:

$$\alpha - \frac{2B}{n+1} - B\epsilon \leq \mathbb{E}[g(Z)] \leq \alpha + B\epsilon. \tag{F.41}$$

□

F.1.7 Proof of Proposition 6.5.3

Proof. It is left to prove that $\tilde{L}_i(\lambda)$ satisfies the conditions of Theorem 6.3.1. It is clear that $\tilde{L}_i(\lambda) \leq 1$ and $\tilde{L}_i(\lambda)$ is non-increasing in λ when $L_i(\lambda)$ is. Since $L_i(\lambda)$ is non-increasing and right-continuous, for any sequence $\lambda_m \downarrow \lambda$,

$$L_i(\lambda_m) \uparrow L_i(\lambda) \Longrightarrow \mathbf{1}\left\{L_i(\lambda_m) > \alpha\right\} \to \mathbf{1}\left\{L_i(\lambda) > \alpha\right\}.$$

Thus, $\tilde{L}_i(\lambda)$ is right-continuous.

Finally, $L_i(\lambda_{\max}) \leq \alpha$ implies $\tilde{L}_i(\lambda_{\max}) = 0 \leq 1 - \beta$. □

F.1.8 Proof of Proposition 6.5.4

Proof. Examining (6.46), for each $\gamma \in \Gamma$, we have

$$\mathbb{E}\left[L(\hat{\lambda}, \gamma)\right] \leq \mathbb{E}\left[L(\hat{\lambda}_\gamma, \gamma)\right] \leq \alpha(\gamma). \tag{F.42}$$

Thus, dividing both sides by $\alpha(\gamma)$ and taking the supremum, we get that

$$\sup_{\gamma \in \Gamma} \mathbb{E}\left[\frac{L(\hat{\lambda}, \gamma)}{\alpha(\gamma)}\right] \leq 1, \tag{F.43}$$

and the worst-case risk is controlled. □

F.1.9 Proof of Proposition 6.5.5

Proof. Because $L_i(\lambda, \gamma)$ is bounded and monotone in λ for all choices of γ, it is also true that $\tilde{L}_i(\lambda)$ is bounded and monotone. Furthermore, the pointwise supremum of right-continuous functions is also right-continuous. Therefore, the $\tilde{L}_i$ satisfy the assumptions of Theorem 6.3.1. □

F.1.10 Proof of Proposition 6.5.6

Proof. Let

$$\hat{\lambda}'_k = \inf\left\{\lambda : \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1,\dots,n+k\}, |\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) \leq \alpha\right\}.$$

Since $L_{\mathcal{S}}(\lambda_{\max}) \le \alpha$, $\hat{\lambda}'_k$ exists almost surely. Since $L_{\mathcal{S}}(\lambda) \le B$, we have

$$\begin{aligned}
&\frac{k!n!}{(n+k)!} \sum_{\mathcal{S}\subset\{1,\ldots,n+k\},|\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) \\
&\le \frac{k!n!}{(n+k)!} \sum_{\mathcal{S}\subset\{1,\ldots,n\},|\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) + B \cdot \sum_{\mathcal{S}\cap\{n+1,\ldots,n+k\}\neq\emptyset,|\mathcal{S}|=k} 1 \\
&= \frac{k!n!}{(n+k)!} \sum_{\mathcal{S}\subset\{1,\ldots,n\},|\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) + B\left(1 - \frac{k!n!}{(n+k)!} \sum_{\mathcal{S}\subset\{1,\ldots,n\},|\mathcal{S}|=k} 1\right) \\
&= \frac{k!n!}{(n+k)!} \sum_{\mathcal{S}\subset\{1,\ldots,n\},|\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) + B\left(1 - \frac{(n!)^2}{(n+k)!(n-k)!}\right).
\end{aligned}$$

Since $L_{\mathcal{S}}(\lambda)$ is non-increasing in λ, we conclude that $\hat{\lambda}'_k \le \hat{\lambda}_k$ if the right-hand side of Eq. (6.50) is not empty; otherwise, by definition, $\hat{\lambda}'_k \le \lambda_{\max} = \hat{\lambda}_k$. Thus, $\hat{\lambda}'_k \le \hat{\lambda}_k$ almost surely. Let E be the multiset of loss functions $\{L_{\mathcal{S}} : \mathcal{S} \subset \{1,\ldots,n+k\}, |\mathcal{S}| = k\}$. Using the same argument in the end of the proof of Theorem 6.3.1 and the right-continuity of $L_{\mathcal{S}}$, we can show that

$$\mathbb{E}\left[L_{\{n+1,\ldots,n+k\}}(\hat{\lambda}'_k) \mid E\right] = \frac{k!n!}{(n+k)!} \sum_{\mathcal{S}\subset\{1,\ldots,n+k\},|\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) \le \alpha.$$

The proof is then completed by the law of iterated expectation. □