

Statistical Guarantees for Black-Box Models

Anastasios Angelopoulos

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-226

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-226.html>

December 20, 2024



Copyright © 2024, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Statistical Guarantees for Black-Box Models

By

Anastasios N. Angelopoulos

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael I. Jordan, Co-chair

Professor Jitendra Malik, Co-chair

Professor Ryan Tibshirani

Professor Laura Waller

Fall 2024

Statistical Guarantees for Black-Box Models

Copyright 2024

by

Anastasios N. Angelopoulos

Abstract

Statistical Guarantees for Black-Box Models

by

Anastasios N. Angelopoulos

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Michael I. Jordan, Co-chair

Professor Jitendra Malik, Co-chair

Reliability has emerged as one of the most important challenges facing AI deployments. One difficulty is the inability of standard theoretical tools to guarantee strong performance of modern AI systems due to their complexity and ever-changing training and development pipelines. Here, we take a different strategy: ensuring reliability of a *black-box* model—one where we only have access to the inputs and outputs, and no knowledge of the mapping between the two. The only way to ensure reliability of such models is by surrounding them with a statistical infrastructure for measurement and calibration.

This thesis develops statistical guarantees for black-box AI models in the domains of prediction and inference. The first part will deal with prediction: guaranteeing reliability on a per-input basis. I will in particular focus on a line of work extending the model-agnostic guarantees of conformal prediction to the realm of risk control and decision-making. The second part will deal with inference, or aggregating predictions to produce estimators, confidence intervals, and p-values to learn about the broader world. There, I will focus on Prediction-Powered Inference, a tool for trustworthy AI-driven science, and its application to automated evaluation of AI algorithms.

To Tassoula, Euripides, Stavros, Dena, Vassilis, Mary, Coco, and Tijana.

Contents

Contents	ii
1 Introduction	1
I Distribution-Free Predictive Guarantees	3
2 Conformal Risk Control	4
2.1 Introduction	4
2.2 Theory	6
2.3 Examples	10
2.4 Extensions	15
2.5 Conclusion	19
2.6 Appendix for Conformal Risk Control	19
3 Conformal Decision Theory	27
3.1 Introduction	27
3.2 Related Work	29
3.3 Conformal Decision Theory	29
3.4 Theory & Conformal Controller Algorithm	30
3.5 Experiments	33
3.6 Discussion & Conclusion	41
II Prediction-Powered Inference	42
4 Core Methodology	43
4.1 Introduction	43
4.2 Main theory: Convex estimation	48
4.3 Applications	55
4.4 Extensions	59
4.5 Appendix for Prediction-Powered Inference	64

5 Using Synthetic Data for Model Evaluation	90
5.1 Introduction	90
5.2 Autoevaluating Accuracy and other Metrics	92
5.3 Evaluating Model Performance from Pairwise Comparisons	100
5.4 Supplementary Material	103
Bibliography	106

Acknowledgments

So many people—collaborators, family, and friends—have shaped my academic journey. It would be impossible to put my feelings into words, or to name all the people to whom I am indebted. Nonetheless, I will try my best.

Thank you to my advisors, Michael Jordan and Jitendra Malik. The two of them have inspired me so much. They believed in me early, and gave me the space to develop my own ideas, even when it was highly unclear that they would amount to anything. I was an atypical student for both Mike and Jitendra—I had taken essentially no formal mathematics or statistics, nor had I exhibited any skill or aptitude in computer vision or deep learning. Hence, as they have both since admitted, I was a risk. Against all odds, Mike and Jitendra saw something in me and took me under their wings. Mike and Jitendra are an unusual choice of advisor combination, and their advice was often conflicting: Mike would often tell me to “eat my vegetables” and focus on contributing solid theory to a core topic in statistical machine learning. Jitendra provided an alternative outlook. After a trip to Almare, he told me the following. “There are two types of researchers: farmers and cowboys. Farmers tend to their research, growing it slowly over many seasons. Cowboys will ride free to find an exciting problem and lasso it. So far, you have been a farmer. It is time to be a cowboy!” I hope that I have managed to make them both happy. All credit goes to them for creating the ideal environment for me to produce the work in this thesis. They have imparted on me a taste for fundamental work, creativity, and ethics that I will take with me forever.

Thank you also to SAIL and the Malik group. I have been inspired by your brilliance, accelerated by your collaboration, and humbled by your friendship. Especially, I would like to thank my cohort-mates, without whom the Ph.D. would have been no fun, and many of whom I count among my closest friends, including Wei-Lin Chiang, Reese Pathak, Ilija Radosavović, Neha Wadia, and Mariel Werner.

Next, I would like to thank my collaborators. I have gotten to learn so much from so many brilliant experts in biology, statistics, medicine, machine learning, computer systems, and more. From SAIL and EMK, I have been privileged to collaborate with and learn from Stephen Bates, Ron Boger, Pierre Boyeau, Tiffany Ding, Lihua Lei, Jordan Lekeufack, Drew Nguyen, Reese Pathak, Mariel Werner, Clara Wong-Fannjiang, Banghua Zhu, and Tijana Zrnić. Special thanks to my longstanding collaborator, Stephen Bates, with whom I have spent countless late nights working over the past five years on our papers and 2 books together. It has been a privilege to learn the foundations of statistical inference from you and to have our paths tied together as they are. I would also like to thank Adam Fisch, without whom conformal risk control would not exist, and who graciously mentored me as his intern at Google this past summer.

I have been lucky also to collaborate with a number of faculty besides my advisors, who remain my close mentors to this day. Mark Humayun was my first mentor, and we did clinical research together. From Mark, I learned to think big and invent, even when it trans-

gresses standard disciplinary boundaries. Gordon Wetzstein, my undergraduate research advisor at Stanford EE, taught me many of the mechanics of research, as well as how to do useful and high-quality experimental work. Laura Waller changed my life by bringing me to Berkeley, and mentored me in all things computational imaging. Her research advice has been indispensable in shaping my intellectual growth from the beginning; in the realm of imaging, she taught me totally new ways of looking at imaging with randomness that have been core to my understanding of measurement as a concept. Ryan Tibshirani came to Berkeley two years ago and our collaborations have been pivotal for me. Together, we have done some of the deepest work of my career, and he serves as an incredible role model of what an academic mentor should be—brilliant, expert, engaged, and above all, thoughtful and kind. Rina Barber also took a chance on me when she agreed to write a book on conformal prediction together. Rina drastically changed the way I approach mathematical statistics, and taught me a level of detail that changed the way I approach my research as well as logical thinking in general. She truly deserves to be called a “genius.” Finally, I would like to thank my newest mentor, Ion Stoica, for all his support and collaboration so far. He and Wei-Lin have taught me so much about how to build great, useful software that makes the world better.

Next, I’d like to thank my closest friends, Amit Kohli and Justin Lewis-Weber. They have been a source of constant fun and support over the years. Amit has been my housemate since our undergraduate years and throughout our doctorates; I aspire to the way he moves through life. It has been incredible working with you, and thank you for teaching me about imaging, research, religion, music, and life. I follow Justin’s guidance on business, food, memes, and just about everything else. He has helped me believe in myself as a leader, and broadened my horizons so greatly.

My family is the most important thing in the world to me. Thank you first to my grandparents. Yaya Tassoula, thank you for giving me your name. I hope that you are proud of how I am treating it. Pappou Euripidi, thank you for encouraging me to live up to my potential. I miss you. Stavros and Dena, thank you for raising me to be the man I am today; I am your image. Mom and dad, without your support I would never have made it this far. Your faith is my strength. Thank you for teaching me hard work, strategic thinking, family values, ethics, and everything else. Coco, you are the best sister I could ask for, thank you for always believing in me. You are always there for me when I need you, and I look up to your taste, your artistry, the way you treat those around you, and more. I love you all so much.

The greatest accomplishment of my Ph.D. has been ending up with the sweetest, most beautiful and brilliant girl in the world. We began as collaborators in Mike’s group, but luckily, she ended up saying yes to going on a date with me. Tijana, you are the love of my life. Every day with you is a blessing, and I wouldn’t be half the man I am today without your kindness, companionship, and guidance. Thank you for being the best imaginable partner in life. Our little family will be our greatest project together!

Chapter 1

Introduction

This thesis is devoted to a single question:

How can we build reliable systems from black-box AI models?

Answering this question is difficult because modern AI models, though undeniably useful, can be wrong in unpredictable ways. These models learn biases and spurious associations that can lead them to make mistakes and output falsehoods. Meanwhile, these mistakes are silent and difficult to debug due to their statistical nature, arising from the data on which the models were trained. Worse yet, models are often black boxes, with the training data and algorithm unknown. But to use AI in critical applications where they could have the most positive impact—from power plants to hospitals, where safety and lives are at stake—we need trust. Therein lies the contradiction: we have to trust AI models that we cannot understand. We need new frameworks for calibrating AI software systems to achieve reliable deployment, despite the fact that the model is a black-box. This problem is on the critical path towards using AI for the greater good.

The key lies in rethinking the way we build larger systems that incorporate AI. Rather than expecting good performance from models, this thesis focuses on building statistical infrastructure that can anticipate and mitigate model failures. The black-box model is treated as a module in an overall system, surrounded by statistical inference procedures that identify errors, quantify uncertainty, and compensate for these in downstream decisions. This infrastructure requires one core ingredient: new statistical methods that assume nothing about the model’s structure, but nonetheless, ensure strong overall system performance, whether that means maintaining safety standards, reducing costs, or obtaining valid scientific conclusions.

The research in this thesis spans theory, experimentation, and deployment. Theoretically, it develops the language of model reliability through advances in the core statistical framework of conformal prediction. The experiments serve to demonstrate the generality of the approach, involve complex learning tasks, from emergency room stroke detection, to

robotic planners and controllers, to object detection, to large language models, to biomedical foundation models, and beyond. Finally, although this thesis focuses on methodology, it is worth mentioning that the techniques have found real use. Through collaborations with hospitals and industry partners in the energy and insurance sectors, these methods have been implemented in the United States at the national scale.

This thesis is based on works co-authored with Stephen Bates, Andrea Bajcsy, Pierre Boyeau, Adam Fisch, Michael I. Jordan, Lihua Lei, Jordan Lekeufack, Jitendra Malik, Tal Schuster, Clara Wong-Fannjiang, Nir Yosef, and Tijana Zrnic. The chapters are each meant to be read independently, and introduce their own mathematical notation due to the spread of topics covered.

Part I

**Distribution-Free Predictive
Guarantees**

Chapter 2

Conformal Risk Control

2.1 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 [120, 156], which we will henceforth refer to simply as “conformal prediction”, has been useful in areas such as computer vision [14] and natural language processing [67] to provide such a guarantee. By measuring the model’s performance on a *calibration dataset* $\{(X_i, Y_i)\}_{i=1}^n$ of feature-response pairs, conformal prediction post-processes the model to construct prediction sets that bound the *miscoverage*,

$$\mathbb{P}(Y_{n+1} \notin \mathcal{C}(X_{n+1})) \leq \alpha, \quad (2.1)$$

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. Note that \mathcal{C} is formed using the first n data points, and the probability in (2.1) 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

$$\mathbb{E}[\ell(\mathcal{C}(X_{n+1}), Y_{n+1})] \leq \alpha, \quad (2.2)$$

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 (2.2) recovers the conformal miscoverage guarantee in (2.1)

when using the miscoverage loss, $\ell(\mathcal{C}(X_{n+1}), Y_{n+1}) = \mathbb{1}\{Y_{n+1} \notin \mathcal{C}(X_{n+1})\}$. However, our algorithm also extends conformal prediction to situations where other loss functions, such as the false negative rate (FNR) or F1-score, are more appropriate.

As an example, consider multilabel classification, where the $Y_i \subseteq \{1, \dots, K\}$ are sets comprising a subset of K classes. Given a trained multilabel classifier $f : \mathcal{X} \rightarrow [0, 1]^K$, 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 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 $\ell(\mathcal{C}_{\hat{\lambda}}(X_{n+1}), Y_{n+1}) = 1 - |Y_{n+1} \cap \mathcal{C}_{\hat{\lambda}}(X_{n+1})|/|Y_{n+1}|$. 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.

2.1.1 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}_λ , 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 $\{(X_i, Y_i)\}_{i=1}^n$ so that risk control as in (2.2) holds.

We now rewrite this same task in a more notationally convenient and abstract form. Consider an exchangeable collection of non-increasing, random functions $L_i : \Lambda \rightarrow (-\infty, B]$, $i = 1, \dots, n + 1$. Throughout the chapter, 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:

$$\mathbb{E}\left[L_{n+1}(\hat{\lambda})\right] \leq \alpha. \quad (2.3)$$

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 (2.3) coincides with risk control as in (2.2).

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

$$\hat{\lambda} = \inf \left\{ \lambda : \frac{n}{n+1} \widehat{R}_n(\lambda) + \frac{B}{n+1} \leq \alpha \right\}. \quad (2.4)$$

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 (2.3). 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,

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

We show the reduction from conformal risk control to conformal prediction in Section 2.2.3. Furthermore, if the risk is non-monotone, then this algorithm does not control the risk; we discuss this in Section 2.2.4. Finally, we provide both practical examples using real-world data and several theoretical extensions of our procedure in Sections 2.3 and 2.4, respectively.

2.1.2 Related work

Conformal prediction was developed by Vladimir Vovk and collaborators beginning in the late 1990s [155, 156], 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. See [13] for a modern introduction to the area or [141] for a more classical alternative. As previously discussed, in this chapter we primarily build on *split conformal prediction* [120]; statistical properties of this algorithm including the coverage upper bound were studied in [100]. Recently there have been many extensions of the conformal algorithm, mainly targeting deviations from exchangeability [19, 64, 72, 146] and improved conditional coverage [14, 18, 75, 132, 133]. Most relevant to us is recent work on risk control in high probability [6, 23, 153] and its applications [11, 12, 68, 122, 136, 138, 139, *inter alia*]. 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 (2.1). The guarantee available through conformal risk control, (2.3), 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 (2.3) 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 [15]). 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, herein we target distribution-free finite-sample control of (2.3), but as a side-note it is also worth pointing the reader to the rich literature on functional central limit theorems [51], which are another way of estimating risk functions.

2.2 Theory

In this section, we establish the core theoretical properties of conformal risk control. All proofs, unless otherwise specified, are deferred to Appendix 2.6.2.

2.2.1 Risk control

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

Theorem 1. *Assume that $L_i(\lambda)$ is non-increasing in λ , right-continuous, and*

$$L_i(\lambda_{\max}) \leq \alpha, \quad \sup_{\lambda} L_i(\lambda) \leq B < \infty \text{ almost surely.} \quad (2.5)$$

Then

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

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

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

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,

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

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 ,

$$\mathbb{E}[L_{n+1}(\hat{\lambda})] \leq \mathbb{E}[L_{n+1}(\hat{\lambda}')]. \quad (2.6)$$

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

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

The proof is completed by the law of total expectation and (2.6). \square

2.2.2 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 :

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

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 2. *In the setting of Theorem 1, further assume that the L_i are i.i.d., $L_i \geq 0$, and for any λ , $\mathbb{P}(J(L_i, \lambda) > 0) = 0$. Then*

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

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

Proposition 3. *In the setting of Theorem 2, for any $\epsilon > 0$, there exists a loss function and $\alpha \in (0, 1)$ such that*

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

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

2.2.3 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 (2.2) specializes to the conformal coverage guarantee in (2.1) 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 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 $s : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. Based on this score function, we create prediction sets for the test point X_{n+1} as

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

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 $[(n+1)(1-\alpha)]/n$ 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* $L_i^{\text{Cvg}}(\lambda) = \mathbb{1}\{Y_i \notin \widehat{\mathcal{C}}_\lambda(X_i)\} = \mathbb{1}\{s(X_i, Y_i) > \lambda\}$. Direct calculation of $\hat{\lambda}$ from (2.4) then

shows the equivalence of the proposed procedure to conformal prediction:

$$\hat{\lambda} = \inf \left\{ \lambda : \frac{1}{n+1} \sum_{i=1}^n \mathbb{1} \{s(X_i, Y_i) > \lambda\} + \frac{1}{n+1} \leq \alpha \right\} =$$

$$\underbrace{\inf \left\{ \lambda : \frac{1}{n} \sum_{i=1}^n \mathbb{1} \{s(X_i, Y_i) \leq \lambda\} \geq \frac{[(n+1)(1-\alpha)]}{n} \right\}}_{\text{conformal prediction algorithm}}.$$

Next, we discuss how the risk lower bound relates to its conformal prediction equivalent. In the setting of conformal prediction, [100] 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 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 2 about the distribution of discontinuities specializes to the continuity of the score function when the miscoverage loss is used:

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

However, the bound for the conformal case is better than the bound for the general case in Theorem 2 by a factor of two, which cannot be improved according to Proposition 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 .

2.2.4 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, (2.3) does not hold. We show this by example.

Proposition 4. *For any ϵ , there exists a non-monotone loss function such that*

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

Notice that for any desired level α , the expectation in (2.3) 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 (2.3), 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 1.

Corollary 5. *Allow $L_i(\lambda)$ to be any (possibly non-monotone) function of λ satisfying 2.5. Take*

$$\tilde{L}_i(\lambda) = \sup_{\lambda' \geq \lambda} L_i(\lambda'), \quad \tilde{R}_n(\lambda) = \frac{1}{n} \sum_{i=1}^n \tilde{L}_i(\lambda) \quad \text{and} \quad \tilde{\lambda} = \inf \left\{ \lambda : \frac{n}{n+1} \tilde{R}_n(\lambda) + \frac{B}{n+1} \leq \alpha \right\}.$$

Then,

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

If the loss function is already monotone, then $\tilde{\lambda}$ reduces to $\hat{\lambda}$. We propose a further algorithm for picking λ in Appendix 2.6.1 that provides an asymptotic risk-control guarantee for *non-monotone* loss functions. However, this algorithm again is only powerful when the risk $\mathbb{E}[L_{n+1}(\lambda)]$ is near-monotone and reduces to the standard conformal risk control algorithm when the loss is monotone.

2.3 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}_λ 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 $\leq \alpha$). Code to reproduce our examples is available at our GitHub (link removed for anonymity).

2.3.1 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 $Y_i \in \wp(\{(1,1), (1,2), \dots, (d,d)\})$, where \wp denotes the power set. We build on an image segmentation model $f : \mathcal{X} \rightarrow [0,1]^{d \times d}$ outputting a probability for each pixel and measure loss as the fraction of false negatives,

$$L_i^{\text{FNR}}(\lambda) = 1 - \frac{|Y_i \cap \mathcal{C}_\lambda(X_i)|}{|Y_i|}, \text{ where } \mathcal{C}_\lambda(X_i) = \{y : f(X_i)_y \geq 1 - \lambda\}. \quad (2.7)$$

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 2.2.1 to pick $\hat{\lambda}$ by (2.4) that controls the FNR on a new point, resulting in the following guarantee:

$$\mathbb{E}[L_{n+1}^{\text{FNR}}(\hat{\lambda})] \leq \alpha. \quad (2.8)$$

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 [63] as our base model f and used $n = 1000$, and evaluated

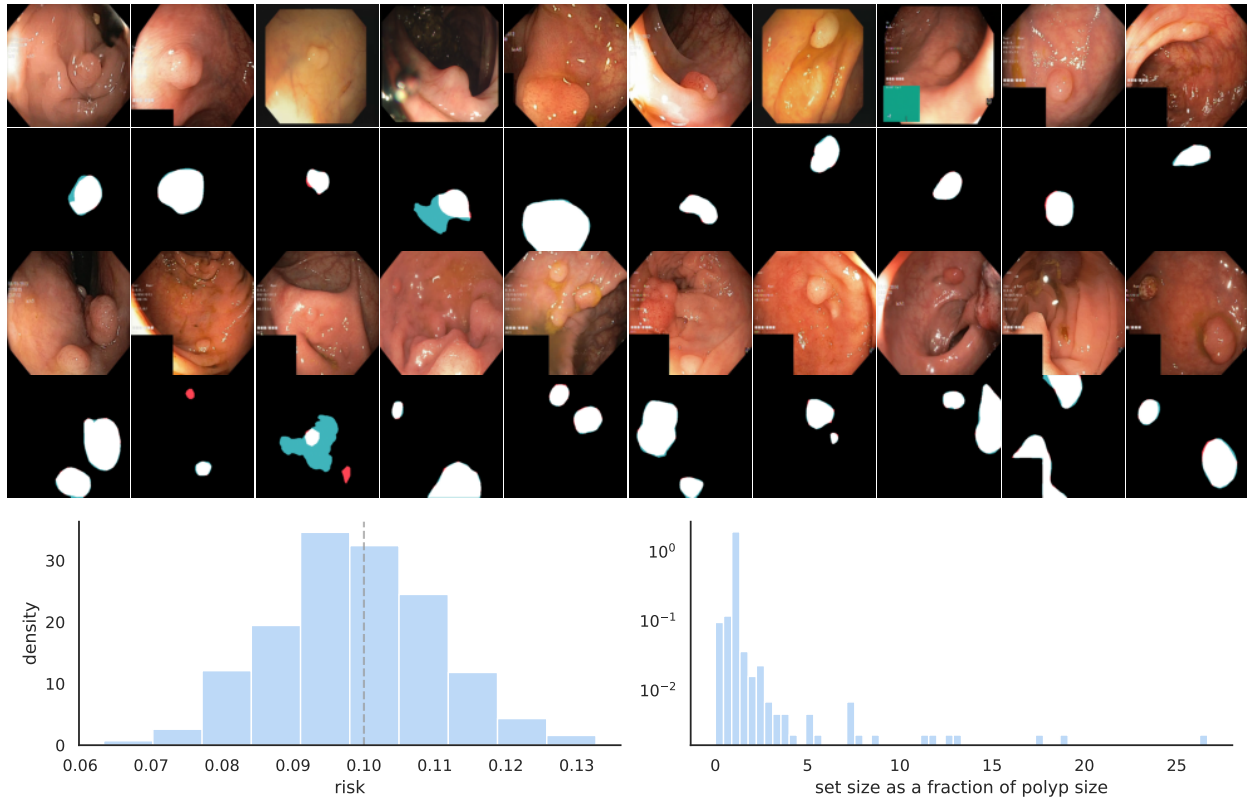


Figure 2.1: **FNR control in tumor segmentation.** The top figure shows examples of our procedure with correct pixels in white, false positives in blue, and false negatives in red. The bottom plots report FNR and set size over 1000 independent random data splits. The dashed gray line marks α .

risk control with the 781 remaining validation data points. We report results with $\alpha = 0.1$ in Figure 2.1. The mean and standard deviation of the risk over 1000 trials are 0.0987 and 0.0114, respectively.

2.3.2 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 $f: \mathcal{X} \rightarrow [0, 1]^K$, we form prediction sets and calculate the number of false positives exactly as in (2.7). By Theorem 1, picking $\hat{\lambda}$ as in (2.4) again yields the FNR-control guarantee in (2.8).

We use the Microsoft Common Objects in Context (MS COCO) computer vision dataset [104], a large-scale 80-class multiclass classification baseline dataset commonly used in computer



Figure 2.2: **FNR control on MS COCO**. The top figure shows examples of our procedure with correct classes in black, 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 .

vision, to evaluate the proposed procedure. We choose a TResNet [127] 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 2.2. The mean and standard deviation of the risk over 1000 trials are 0.0996 and 0.0052, respectively.

2.3.3 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 $Y_i \in \{1, \dots, K\}$ on a tree with nodes \mathcal{V} and edges \mathcal{E} . Using a single-class classification model $f : \mathcal{X} \rightarrow \Delta^K$, 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 $d : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{Z}$ be the function that

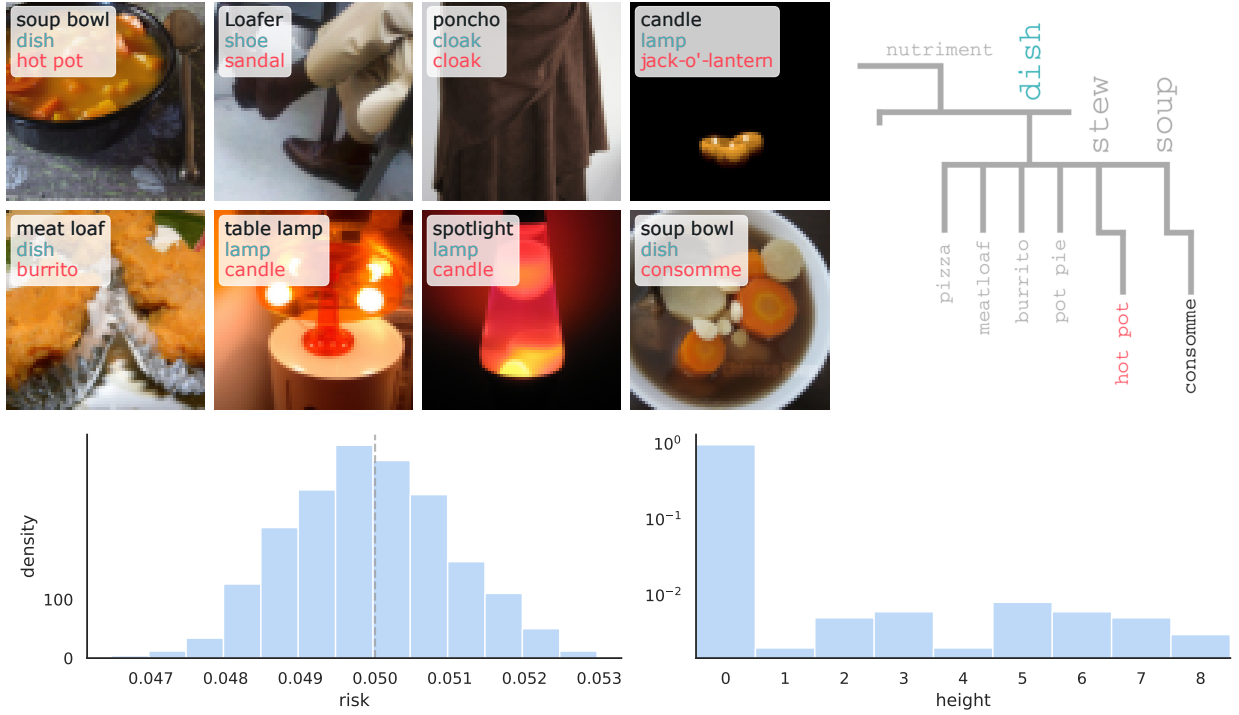


Figure 2.3: **Control of graph distance on hierarchical ImageNet.** The top figure shows examples of our procedure with correct classes in black, false positives in blue, and false negatives in red. The bottom plots report our minimum hierarchical distance loss and set size over 1000 independent random data splits. The dashed gray line marks α .

returns the length of the shortest path between two nodes, let $\mathcal{A} : \mathcal{V} \rightarrow 2^{\mathcal{V}}$ be the function that returns the ancestors of its argument, and let $\mathcal{P} : \mathcal{V} \rightarrow 2^{\mathcal{V}}$ be the function that returns the set of leaf nodes that are descendants of its argument. We also let $g(v, x) = \sum_{k \in \mathcal{P}(v)} f(x)_k$ be the sum of scores of leaves descended from v . Further, define a hierarchical distance

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

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

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

This loss returns zero if y is a child of any element in \mathcal{C}_λ , and otherwise returns the minimum distance between any element of \mathcal{C}_λ 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 (2.4) to achieve the guarantee

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

For this experiment, we use the ImageNet dataset [55], which comes with an existing label hierarchy, WordNet, of maximum depth $D = 14$. We choose a ResNet152 [80] as our base model f and used $n = 30000$, and evaluated risk control with the remaining 20000. We report results with $\alpha = 0.05$ in Figure 2.3. The mean and standard deviation of the risk over 1000 trials are 0.0499 and 0.0011, respectively.

2.3.4 F1-score control in open-domain question answering

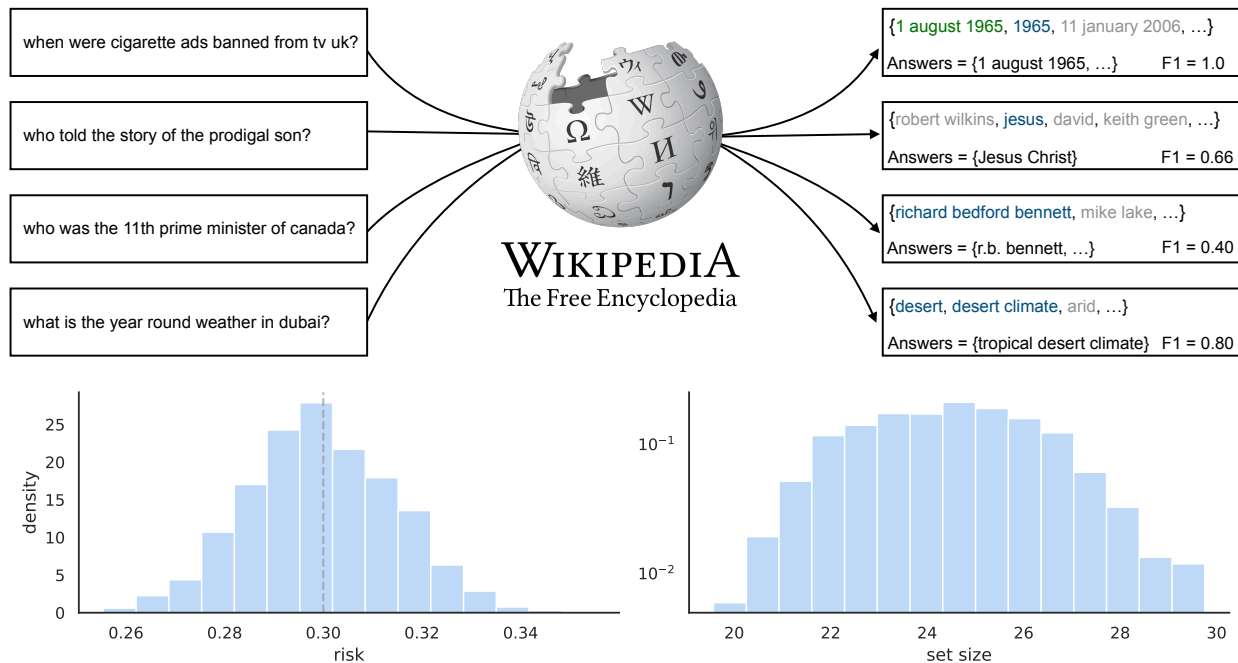


Figure 2.4: **F1-score control on Natural Questions.** The top figure shows examples of our procedure with fully correct answers in green, partially correct answers in blue, and false positives in gray. Note that due to the nature of the evaluation, answers that are technically correct may still be down-graded 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 .

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

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

Formally, 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 $f: \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$, we calibrate the *best* token-based F1-score of the prediction set, taken over all pairs of predictions and answers:

$$L_i^{\text{F1}}(\lambda) = 1 - \max \{ \text{F1}(a, c) : c \in \mathcal{C}_\lambda(X_i), a \in Y_i \},$$

$$\text{where } \mathcal{C}_\lambda(X_i) = \{ y \in \mathcal{V}^m : f(X_i, y) \geq \lambda \}.$$

We define the F1-score following popular QA evaluation metrics [125], 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 2.2.1 to achieve the following guarantee:

$$\mathbb{E} [L_{n+1}^{\text{F1}}(\hat{\lambda})] \leq \alpha.$$

We use the Natural Questions (NQ) dataset [94], 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 [91]. Our base model is the DPR Retriever-Reader model [91], 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 2.4. The mean and standard deviation of the risk over 1000 trials are 0.2996 and 0.0150, respectively.

2.4 Extensions

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

2.4.1 Risk control under distributional shift

Suppose the researcher wants to control the risk under a distribution shift. Then the goal in (2.3) can be redefined as

$$\mathbb{E}_{(X_1, Y_1), \dots, (X_n, Y_n) \sim P_{\text{train}}, (X_{n+1}, Y_{n+1}) \sim P_{\text{test}}} [L_{n+1}(\hat{\lambda})] \leq \alpha, \quad (2.9)$$

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 (2.9) can be rewritten as

$$\mathbb{E}_{(X_1, Y_1), \dots, (X_{n+1}, Y_{n+1}) \sim P_{\text{train}}} \left[w(X_{n+1}, Y_{n+1}) L_{n+1}(\hat{\lambda}) \right] \leq \alpha, \quad (2.10)$$

where $w(x, y) = \frac{dP_{\text{test}}(x, y)}{dP_{\text{train}}(x, y)}$.

When w is known and bounded, we can apply our procedure on the loss function $\tilde{L}_{n+1}(\lambda) = w(X_{n+1}, Y_{n+1}) L_{n+1}(\lambda)$, which is non-decreasing, bounded, and right-continuous in λ whenever L_{n+1} is. Thus, Theorem 1 guarantees that the resulting $\hat{\lambda}$ satisfies (2.10).

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

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

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

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

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. *In the setting of Theorem 1,*

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

It is easy to show that the weighted conformal procedure [146] is a special case with $L_i(\lambda) = \mathbb{1}\{Y_i \notin \mathcal{C}_\lambda(X_i)\}$ where $\mathcal{C}_\lambda(X_i)$ is the prediction set that thresholds the conformity score at λ . Thus, Proposition 6 generalizes [146] 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 [101] and [35] 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 [19] 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 $\hat{\lambda}(Z_1, \dots, Z_n)$ to refer to that chosen in (2.4).

Proposition 7. *Let $Z = (Z_1, \dots, Z_{n+1})$ be a sequence of random variables. Then, under the conditions in Theorem 1,*

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

If further the assumptions of Theorem 2 hold,

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

2.4.2 Quantile risk control

[143] generalizes [23] 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

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

Analogous to (2.3), we want to find $\hat{\lambda}$ based on $(X_i, Y_i)_{i=1}^n$ such that

$$\text{Quantile}_\beta(L_{n+1}(\hat{\lambda}_\beta)) \leq \alpha. \tag{2.11}$$

By definition,

$$\text{Quantile}_\beta(L_{n+1}(\hat{\lambda}_\beta)) \leq \alpha \iff \mathbb{E}[\mathbb{1}\{L_{n+1}(\hat{\lambda}_\beta) > \alpha\}] \leq 1 - \beta.$$

As a consequence, quantile risk control is equivalent to expected risk control (2.3) with loss function $\tilde{L}_i(\lambda) = \mathbb{1}\{L_i(\lambda) > \alpha\}$. Let

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

Proposition 8. *In the setting of Theorem 1, (2.11) is achieved.*

[143] considers the high-probability control of a wider class of quantile-based risks which include the conditional value-at-risk (CVaR). It is unclear whether those more general risks can be controlled unconditionally. We leave this open problem for future research.

2.4.3 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

$$\sup_{\gamma \in \Gamma} \mathbb{E} \left[\frac{L_i(\hat{\lambda}; \gamma)}{\alpha(\gamma)} \right] \leq 1. \quad (2.12)$$

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

$$\hat{\lambda} = \sup_{\gamma \in \Gamma} \hat{\lambda}_\gamma, \text{ where } \hat{\lambda}_\gamma = \inf \left\{ \lambda : \frac{1}{n+1} \sum_{i=1}^n L_i(\lambda; \gamma) + \frac{B}{n+1} \leq \alpha(\gamma) \right\}. \quad (2.13)$$

Then the risk is controlled.

Proposition 9. *In the setting of Theorem 1, (2.12) is satisfied.*

2.4.4 Adversarial risks

We next show how to control risks defined by adversarial perturbations. We adopt the same notation as Section 2.4.3. [23] (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 an $\hat{\lambda}$ based on $(X_i, Y_i)_{i=1}^n$ such that

$$\mathbb{E}[\sup_{\gamma \in \Gamma} L_i(\lambda; \gamma)] \leq \alpha. \quad (2.14)$$

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:

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

Proposition 10. *In the setting of Theorem 1, (2.14) is satisfied.*

2.4.5 U-risk control

For ranking and metric learning, [23] considered loss functions that depend on two test points. In general, for any $k > 1$ and subset $\mathcal{S} \subset \{1, \dots, n+k\}$ 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

$$\mathbb{E} [L_{\{n+1, \dots, n+k\}}(\hat{\lambda}_k)] \leq \alpha. \quad (2.15)$$

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

$$\hat{\lambda}_k = \inf \left\{ \lambda : \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1, \dots, n\}, |\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) + B \left(1 - \frac{(n!)^2}{(n+k)!(n-k)!} \right) \leq \alpha \right\}. \quad (2.16)$$

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

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

Then (2.15) is achieved.

2.5 Conclusion

This generalization of conformal prediction broadens its scope to new applications, as shown in Section 2.3. The mathematical tools developed in Section 2.2, Section 2.4, 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.

2.6 Appendix for Conformal Risk Control

2.6.1 Monotonizing non-monotone risks

We next show that the proposed algorithm leads to asymptotic risk control for non-monotone risk functions when applied to a monotonized version of the empirical risk. We set the *monotonized empirical risk* to be

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

then define

$$\hat{\lambda}_n^\dagger = \inf \{ \lambda : \widehat{R}_n^\dagger(\lambda) \leq \alpha \}.$$

Theorem 12. *Let the $L_i(\lambda)$ be right-continuous, i.i.d., bounded (both above and below) functions satisfying (2.5). Then,*

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

Theorem 12 implies that an analogous procedure to 2.4 also controls the risk asymptotically. In particular, taking

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

also results in asymptotic risk control (to see this, plug $\tilde{\lambda}^\dagger$ into Theorem 12 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, $\tilde{\lambda}^\dagger = \hat{\lambda}$. However, the counterexample in Proposition 4 does not apply to $\tilde{\lambda}^\dagger$, and it is currently unknown whether this procedure does or does not provide finite-sample risk control.

2.6.2 Proofs

The proof of Theorem 2 uses the following lemma on the approximate continuity of the empirical risk.

Lemma 12.1 (Jump Lemma). *In the setting of Theorem 2, any jumps in the empirical risk are bounded, i.e.,*

$$\sup_{\lambda} J(\widehat{R}_n, \lambda) \stackrel{a.s.}{\leq} \frac{B}{n}.$$

Proof of Jump Lemma, Lemma 12.1. By boundedness, the maximum contribution of any single point to the jump is $\frac{B}{n}$, so

$$\exists \lambda : J(\widehat{R}_n, \lambda) > \frac{B}{n} \implies \exists \lambda : J(L_i, \lambda) > 0 \text{ and } J(L_j, \lambda) > 0 \text{ for some } i \neq j.$$

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

$$\mathbb{P} \left(\exists \lambda : J(\widehat{R}_n, \lambda) > \frac{B}{n} \right) \leq \sum_{i \neq j} \mathbb{P}(\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset)$$

Rewriting each term of the right-hand side using tower property and law of total probability gives

$$\begin{aligned} \mathbb{P}(\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset) &= \mathbb{E} \left[\mathbb{P}(\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset \mid \mathcal{D}_j) \right] \\ &\leq \mathbb{E} \left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}(\lambda \in \mathcal{D}_i \mid \mathcal{D}_j) \right] = \mathbb{E} \left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}(\lambda \in \mathcal{D}_i) \right], \end{aligned}$$

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 $\mathbb{P}(J(L_i, \lambda) > 0) = 0$,

$$\mathbb{E} \left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}(\lambda \in \mathcal{D}_i) \right] = \mathbb{E} \left[\sum_{\lambda \in \mathcal{D}_j} \mathbb{P}(J(L_i, \lambda) > 0) \right] = 0.$$

Chaining the above inequalities yields $\mathbb{P}(\exists \lambda : J(\widehat{R}_n, \lambda) > \frac{B}{n}) \leq 0$, so $\mathbb{P}(\exists \lambda : J(\widehat{R}_n, \lambda) > \frac{B}{n}) = 0$. \square

Proof of Theorem 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

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

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 ,

$$\mathbb{E}[L_{n+1}(\hat{\lambda}'')] \leq \mathbb{E}[L_{n+1}(\hat{\lambda})]$$

By exchangeability and the fact that $\hat{\lambda}''$ is a symmetric function of L_1, \dots, L_{n+1} ,

$$\mathbb{E}[L_{n+1}(\hat{\lambda}'')] = \mathbb{E}[\widehat{R}_{n+1}(\hat{\lambda}'')]$$

For the remainder of the proof we focus on lower-bounding $\widehat{R}_{n+1}(\hat{\lambda}'')$. We begin with the following identity:

$$\alpha = \widehat{R}_{n+1}(\hat{\lambda}'') + \frac{B}{n+1} - \left(\widehat{R}_{n+1}(\hat{\lambda}'') + \frac{B}{n+1} - \alpha \right).$$

Rearranging the identity,

$$\widehat{R}_{n+1}(\hat{\lambda}'') = \alpha - \frac{B}{n+1} + \left(\widehat{R}_{n+1}(\hat{\lambda}'') + \frac{B}{n+1} - \alpha \right).$$

Using the Jump Lemma to bound $\left(\widehat{R}_{n+1}(\hat{\lambda}'') + \frac{B}{n+1} - \alpha \right)$ below by $-\frac{B}{n+1}$ gives

$$\widehat{R}_{n+1}(\hat{\lambda}'') \geq \alpha - \frac{2B}{n+1}.$$

Finally, chaining together the above inequalities,

$$\mathbb{E} \left[L_{n+1}(\hat{\lambda}) \right] \geq \mathbb{E} \left[\widehat{R}_{n+1}(\hat{\lambda}'') \right] \geq \alpha - \frac{2B}{n+1}.$$

\square

Proof of Proposition 3. Without loss of generality, assume $B = 1$. Fix any $\epsilon' > 0$. Consider the following loss functions, which satisfy the conditions in Theorem 2:

$$L_i(\lambda) \stackrel{i.i.d.}{\sim} \begin{cases} 1 & \lambda \in [0, Z_i) \\ \frac{k}{k+1} & \lambda \in [Z_i, W_i) \\ 0 & \text{else} \end{cases},$$

where $k \in \mathbb{N}$, the $Z_i \stackrel{i.i.d.}{\sim} \text{Uniform}(0, 0.5)$, the $W_i \stackrel{i.i.d.}{\sim} \text{Uniform}(0.5, 1)$ for $i \in \{1, \dots, n+1\}$ and $\alpha = \frac{k+1-\epsilon'}{n+1}$. Then, by the definition of $\hat{\lambda}$, we know

$$\widehat{R}_n(\hat{\lambda}) \leq \frac{k-\epsilon'}{n}. \quad (2.17)$$

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 (2.17), we know that $|\{i \in \{1, \dots, n\} : L_i(\hat{\lambda}) > 0\}| \leq \lfloor (k+1)(k-\epsilon')/k \rfloor \leq k$. This immediately implies that

$$\hat{\lambda} \geq W_{(n-k+1)},$$

where $W_{(j)}$ denotes the j -th order statistic. Notice that for all $\lambda > \frac{1}{2}$,

$$R(\lambda) = \mathbb{E}[L_i(\lambda)] = \frac{k}{k+1} \mathbb{P}(W_i > \lambda) = \frac{k}{k+1} \cdot 2(1-\lambda),$$

so $R(\hat{\lambda}) \leq \frac{k}{k+1} \cdot 2(1-W_{(n-k+1)})$. 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, $2(1-W_{(n-k+1)}) \stackrel{d}{=} U_{(k)}$,

$$R(\hat{\lambda}) \leq \frac{k}{k+1} U_{(k)},$$

where \leq denotes the stochastic dominance. It is well-known that $U_{(k)} \sim \text{Beta}(k, n+1-k)$ and hence

$$\mathbb{E}[R(\hat{\lambda})] \leq \frac{k}{k+1} \cdot \frac{k}{n+1}.$$

Thus,

$$\alpha - \mathbb{E}[R(\hat{\lambda})] \geq \frac{k+1-\epsilon}{n+1} - \frac{k^2}{(n+1)(k+1)} = \frac{1}{n+1} \cdot \frac{(2-\epsilon')k+1-\epsilon'}{k+1}.$$

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

$$\alpha - \mathbb{E}[R(\hat{\lambda})] \geq \frac{2-\epsilon}{n+1}.$$

□

Proof of Proposition 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

$$L_i(N/N) = 0, \quad (L_i(0/N), L_i(1/N), \dots, L_i((N-1)/N)) \stackrel{i.i.d.}{\sim} \text{Ber}(p).$$

By definition, $\hat{\lambda}$ is independent of L_{n+1} . Thus, for any $j = 0, 1, \dots, N-1$,

$$\{L_{n+1}(\hat{\lambda}) \mid \hat{\lambda} = j/N\} \sim \text{Ber}(p), \quad \{L_{n+1}(\hat{\lambda}) \mid \hat{\lambda} = 1\} \sim \delta_0.$$

Then,

$$\mathbb{E}[L_{n+1}(\hat{\lambda})] = p \cdot \mathbb{P}(\hat{\lambda} \neq 1)$$

Note that

$$\hat{\lambda} \neq 1 \iff \min_{j \in \{0, \dots, N-1\}} \frac{1}{n+1} \sum_{i=1}^n L_i(j/N) \leq \alpha - \frac{1}{n+1}.$$

Since $\alpha > 1/(n+1)$,

$$\begin{aligned} \mathbb{P}(\hat{\lambda} \neq 1) &= 1 - \mathbb{P}(\hat{\lambda} = 1) = 1 - \mathbb{P}\left(\text{for all } j, \text{ we have } \frac{1}{n+1} \sum_{i=1}^n L_i(j/N) > \alpha - \frac{1}{n+1}\right) \\ &= 1 - \left(\sum_{k=\lceil (n+1)\alpha \rceil}^n \binom{n}{k} p^k (1-p)^{(n-k)}\right)^N \\ &= 1 - \left(1 - \text{BinoCDF}(n, p, \lceil (n+1)\alpha \rceil - 1)\right)^N \end{aligned}$$

As a result,

$$\mathbb{E}[L_{n+1}(\hat{\lambda})] = p \left(1 - \left(1 - \text{BinoCDF}(n, p, \lceil (n+1)\alpha \rceil - 1)\right)^N\right).$$

Now let N be sufficiently large such that

$$\left(1 - \left(1 - \text{BinoCDF}(n, p, \lceil (n+1)\alpha \rceil - 1)\right)^N\right) > p.$$

Then

$$\mathbb{E}[L_{n+1}(\hat{\lambda})] > p^2$$

For any $\alpha > 0$, we can take p close enough to 1 to render the claim false. \square

Proof of Theorem 12. Define the *monotonized population risk* as

$$R^\dagger(\lambda) = \sup_{t \geq \lambda} \mathbb{E}[L_{n+1}(t)]$$

Note that the independence of L_{n+1} and $\hat{\lambda}_n^\dagger$ implies that for all n ,

$$\mathbb{E}[L_{n+1}(\hat{\lambda}_n^\dagger)] \leq \mathbb{E}[R^\dagger(\hat{\lambda}_n^\dagger)].$$

Since R^\dagger is bounded, monotone, and one-dimensional, a generalization of the Glivenko-Cantelli Theorem given in Theorem 1 of [54] gives that uniformly over λ ,

$$\limsup_{n \rightarrow \infty} \sup_{\lambda} |\widehat{R}_n(\lambda) - R(\lambda)| \xrightarrow{a.s.} 0.$$

As a result,

$$\limsup_{n \rightarrow \infty} \sup_{\lambda} |\widehat{R}_n^\dagger(\lambda) - R^\dagger(\lambda)| \xrightarrow{a.s.} 0,$$

which implies that

$$\lim_{n \rightarrow \infty} |\widehat{R}_n^\dagger(\hat{\lambda}^\dagger) - R^\dagger(\hat{\lambda}^\dagger)| \xrightarrow{a.s.} 0.$$

By definition, $\widehat{R}^\dagger(\hat{\lambda}^\dagger) \leq \alpha$ almost surely and thus this directly implies

$$\limsup_{n \rightarrow \infty} R^\dagger(\hat{\lambda}_n^\dagger) \leq \alpha \quad \text{a.s..}$$

Finally, since for all n , $R^\dagger(\hat{\lambda}_n^\dagger) \leq B$, by Fatou's lemma,

$$\lim_{n \rightarrow \infty} \mathbb{E} \left[L_{n+1}(\hat{\lambda}_n^\dagger) \right] \leq \limsup_{n \rightarrow \infty} \mathbb{E} \left[R^\dagger(\hat{\lambda}_n^\dagger) \right] \leq \mathbb{E} \left[\limsup_{n \rightarrow \infty} R^\dagger(\hat{\lambda}_n^\dagger) \right] \leq \alpha.$$

□

Proposition 6. Let

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

Since $\inf_{\lambda} L_i(\lambda) \leq \alpha$, $\hat{\lambda}'$ exists almost surely. Using the same argument as in the proof of Theorem 1, we can show that $\hat{\lambda}' \leq \hat{\lambda}(X_{n+1})$. Since $L_{n+1}(\lambda)$ is non-increasing in λ ,

$$\mathbb{E}[L_{n+1}(\hat{\lambda}(X_{n+1}))] \leq \mathbb{E}[L_{n+1}(\hat{\lambda}')].$$

Let E be the multiset of loss functions $\{(X_1, Y_1), \dots, (X_{n+1}, Y_{n+1})\}$. Then $\hat{\lambda}'$ is a function of E , or, equivalently, $\hat{\lambda}'$ is a constant conditional on E . Lemma 3 of [146] implies that

$$(X_{n+1}, Y_{n+1}) | E \sim \sum_{i=1}^{n+1} \frac{w(X_i)}{\sum_{j=1}^{n+1} w(X_j)} \delta_{(X_j, Y_j)} \implies L_{n+1} | E \sim \sum_{i=1}^{n+1} \frac{w(X_i)}{\sum_{j=1}^{n+1} w(X_j)} \delta_{L_i}$$

where δ_z denotes the Dirac measure at z . Together with the right-continuity of L_i , the above result implies

$$\mathbb{E}[L_{n+1}(\hat{\lambda}') | E] = \frac{\sum_{i=1}^{n+1} w(X_i) L_i(\hat{\lambda}')}{\sum_{i=1}^{n+1} w(X_i)} \leq \alpha.$$

The proof is then completed by the law of total expectation. □

Proposition 7. Define the vector $Z' = (Z'_1, \dots, Z'_n, Z_{n+1})$, where $Z'_i \stackrel{i.i.d.}{\sim} \mathcal{L}(Z_{n+1})$ for all $i \in [n]$. Let

$$\epsilon = \sum_{i=1}^n \text{TV}(Z_i, Z'_i).$$

By sublinearity,

$$\text{TV}(Z, Z') \leq \epsilon. \tag{2.18}$$

It is a standard fact that (2.18) implies

$$\sup_{f \in \mathcal{F}_1} |\mathbb{E}[f(Z)] - \mathbb{E}[f(Z')]| \leq \epsilon,$$

where $\mathcal{F}_1 = \{f : \mathcal{Z} \mapsto [0, 1]\}$. Let $\ell : \mathcal{Z} \times \Lambda \rightarrow [0, B]$ be a bounded loss function. Furthermore, let $g(z) = \ell(z_{n+1}; \hat{\lambda}(z_1, \dots, z_n))$. Since $g(Z) \in [0, B]$,

$$|\mathbb{E}[g(Z)] - \mathbb{E}[g(Z')]| \leq B\epsilon.$$

Furthermore, since Z'_1, \dots, Z'_{n+1} are exchangeable, we can apply Theorems 1 and 2 to $\mathbb{E}[g(Z')]$, recovering

$$\alpha - \frac{2B}{n+1} \leq \mathbb{E}[g(Z')] \leq \alpha.$$

A final step of triangle inequality implies the result:

$$\alpha - \frac{2B}{n+1} - B\epsilon \leq \mathbb{E}[g(Z)] \leq \alpha + B\epsilon.$$

□

Proposition 8. It is left to prove that $\tilde{L}_i(\lambda)$ satisfies the conditions of Theorem 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 $\lambda_m \downarrow \lambda$,

$$L_i(\lambda_m) \uparrow L_i(\lambda) \implies \mathbb{1}\{L_i(\lambda_m) > \alpha\} \rightarrow \mathbb{1}\{L_i(\lambda) > \alpha\}.$$

Thus, $\tilde{L}_i(\lambda)$ is right-continuous. Finally, $L_i(\lambda_{\max}) \leq \alpha$ implies $\tilde{L}_i(\lambda_{\max}) = 0 \leq 1 - \beta$. □

Proposition 9. Examining (2.13), for each $\gamma \in \Gamma$, we have

$$\mathbb{E}[L(\hat{\lambda}, \gamma)] \leq \mathbb{E}[L(\hat{\lambda}_\gamma, \gamma)] \leq \alpha(\gamma).$$

Thus, dividing both sides by $\alpha(\gamma)$ and taking the supremum, we get that $\sup_{\gamma \in \Gamma} \mathbb{E}\left[\frac{L(\hat{\lambda}, \gamma)}{\alpha(\gamma)}\right] \leq 1$, and the worst-case risk is controlled. □

Proposition 10. 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 1. □

Proposition 11. 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 $L_{\mathcal{S}}(\lambda_{\max}) \leq \alpha$, $\hat{\lambda}'_k$ exists almost surely. Since $L_{\mathcal{S}}(\lambda) \leq B$, we have

$$\begin{aligned} & \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1, \dots, n+k\}, |\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) \\ & \leq \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1, \dots, n\}, |\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) + B \cdot \sum_{\mathcal{S} \cap \{n+1, \dots, n+k\} \neq \emptyset, |\mathcal{S}|=k} 1 \\ & = \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1, \dots, n\}, |\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) + B \left(1 - \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1, \dots, n\}, |\mathcal{S}|=k} 1 \right) \\ & = \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1, \dots, n\}, |\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) + B \left(1 - \frac{(n!)^2}{(n+k)!(n-k)!} \right). \end{aligned}$$

Since $L_{\mathcal{S}}(\lambda)$ is non-increasing in λ , we conclude that $\hat{\lambda}'_k \leq \hat{\lambda}_k$ if the right-hand side of (2.16) is not empty; otherwise, by definition, $\hat{\lambda}'_k \leq \lambda_{\max} = \hat{\lambda}_k$. Thus, $\hat{\lambda}'_k \leq \hat{\lambda}_k$ almost surely. Let E be the multiset of loss functions $\{L_{\mathcal{S}} : \mathcal{S} \subset \{1, \dots, n+k\}, |\mathcal{S}|=k\}$. Using the same argument in the end of the proof of Theorem 1 and the right-continuity of $L_{\mathcal{S}}$, we can show that

$$\mathbb{E} \left[L_{\{n+1, \dots, n+k\}}(\hat{\lambda}'_k) \mid E \right] = \frac{k!n!}{(n+k)!} \sum_{\mathcal{S} \subset \{1, \dots, n+k\}, |\mathcal{S}|=k} L_{\mathcal{S}}(\lambda) \leq \alpha.$$

The proof is then completed by the law of iterated expectation. □

Chapter 3

Conformal Decision Theory

3.1 Introduction

Autonomous systems increasingly rely on complex learned models to supply predictions that are the basis for decision-making. Self-driving cars rely on deep neural networks [3, 87, 135, 151] to plan paths around nearby pedestrians, robotic manipulators leverage learned grasp models [111] to plan high-throughput pick-and-place maneuvers in factories, and AI-enabled trading agents optimize the financial future of investors [169]. There is a conceptual gap between prediction and decision-making, and it remains a challenge to ensure that systems make *good decisions* despite *imperfect predictions*.

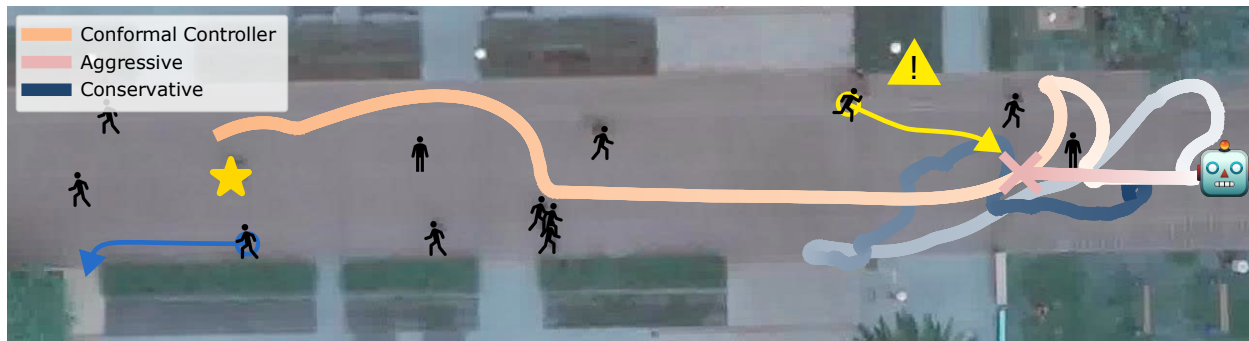


Figure 3.1: Robot planner using a conformal controller on the Stanford Drone Dataset [129]. The future trajectories of humans are predicted online by a machine learning algorithm (not visualized). The robot planner finds an optimal spline through the scene and is penalized for being close to humans. This penalty is proportional to a conformal control variable, λ_t , which is adjusted online by the conformal controller so the average distance from a human is no less than two meters. The orange, red, and blue curves are the robot trajectory with different planners: the conformal controller, an aggressive planner with $\lambda = 0$ (i.e., no reward for avoiding humans), and a conservative planner with a large negative value of λ (i.e., a large reward for avoiding humans). The darkness of the lines indicates the passage of time. Illustrative pedestrian trajectories are plotted as arrows; only the yellow pedestrians affect the spline planner. Details in Section 3.5.1 and videos on [project website](#)[†].

One increasingly popular strategy is to quantify the uncertainty in the predictions independently of their downstream effect on the decision via conformal prediction (CP) [4, 66, 73, 154, 156, 172]. This approach has become popular because, when used to provide simultaneous prediction sets on all outcomes, conformal prediction provides statistical guarantees of safe autonomous behavior without any assumption on the underlying distribution or model. This application of CP has shown impact in robot navigation [48, 58, 105, 115], early warning systems [110], out-of-distribution detection [33, 142], probabilistic pose estimation [168], and for large language models [126]. However, the requirement of simultaneous coverage is challenging to satisfy and for many decision systems is excessive. What if we could provide statistical guarantees, as in CP, *directly* on our decisions, bypassing the need to construct prediction sets?

This chapter presents Conformal Decision Theory, a theoretical and algorithmic framework that unifies predictive uncertainty and safe decision-making. Our key idea is

*instead of calibrating **prediction sets** for coverage, we directly calibrate **decisions** for low risk.*

Our main algorithmic innovation is a class of algorithms called *conformal controllers*. A conformal controller starts with a conformal control variable, λ_t , which determines the decision-maker’s conservatism or aggressiveness. Then, it dynamically adjusts λ_t to balance risk and performance in such a way that guarantees a low risk. The main practical benefit of this approach is its *emergent ability to ignore irrelevant uncertainty*, only accounting for that which *affects decisions*. This can be much less conservative than the prediction-set strategy. For example, in Figure 3.1, the planner only considers the humans that pose a collision risk.

The contributions of this chapter are threefold:

- We introduce Conformal Decision Theory, the idea of directly calibrating decisions with conformal controllers. This extends the line of work in online adversarial conformal prediction [5, 22, 66, 73] to the decision-making setting.
- We prove finite-time risk bounds for conformal controllers. Even when applied to prediction sets, these results are stronger than any previously known results for online adversarial conformal prediction.
- We show the utility of the framework in three simulations where Conformal Decision Theory is applied to robot navigation: the Stanford Drone Dataset [129], a stock trading simulation, and a robot manufacturing example.

The main potential impact of this chapter is to broaden the scope of conformal prediction. Our methods are more appropriate for disciplines that focus on decision-making, such as control theory, reinforcement learning, and logistics. In these disciplines, algorithms are

ultimately evaluated by the decisions, not the predictions, that they make. Furthermore, there are many settings where it does not make sense to construct prediction sets, and our technique can provide a distribution-free outlook for such problems (see, e.g., Section 3.5.2).

3.2 Related Work

Decision-Making Under Predictive Uncertainty. Within the machine learning and statistics community, uncertainty quantification of prediction models has been studied widely, from conformal prediction to Bayesian neural networks to ensembles [70, 74, 93, 96, 97]. Instead of focusing on prediction calibration alone, the controls and optimization community have coupled prediction uncertainty with safe (i.e., risk bounded) decision-making via chance-constrained optimal control [31, 59] and scenario optimization [34, 53]. The former typically constructs prediction sets that are used as constraints while the latter safeguards against samples drawn from the prediction model. Instead of directly calibrating the output of upstream prediction modules or solving decision-making problems under probabilistic constraints, this chapter presents a theoretical and algorithmic approach to tuning the robot’s decision risk directly as a function of historical decision-making performance.

Online Learning & Nonstochastic Control. The method herein is reminiscent of online learning, and specifically the online gradient descent (OGD) update of [179]. The connection is most apparent when examining the forthcoming Equation (3.2) with $\ell_t = \mathbf{1}\{y_t \in \mathcal{C}(x_t)\}$; this recovers the ACI algorithm of [73], which is OGD on the quantile loss [93]. However, the update in (3.2) is substantially more general because it incorporates arbitrary decision rules, and reframing it as OGD on an analytic loss function is generally impossible. Furthermore, the guarantees in [73] are not to our knowledge recoverable by existing regret analyses from online convex optimization and nonstochastic control, e.g., [29, 78, 79]. However, the guarantees do share a retrospective flavor, in that, like regret analyses, they provide guarantees on average over the observed history.

3.3 Conformal Decision Theory

Conformal Decision Theory (CDT) is an approach for calibrating an agent’s decisions to achieve statistical guarantees for the realized average loss of those decisions. Consider a decision-making agent whose input space is \mathcal{X} and action space is \mathcal{U} . In our running example of robot navigation, $x_t \in \mathcal{X}$ captures the current state of the robot, the current scene information (e.g., environment geometry), and the agent information (e.g., pedestrian predictions) while $u_t \in \mathcal{U}$ is the action that the ego vehicle plans at the current time t . At time t , the agent has access to a family of *decision functions*

$$\mathcal{D}_t := \{D_t^\lambda : \mathcal{X} \rightarrow \mathcal{U}, \lambda \in \mathbb{R}\},$$

parameterized by λ , which we call a *conformal control variable*. One should think of λ as indexing the decisions from least to most conservative. In Figure 3.1, \mathcal{D}_t is the set of dynamically feasible splines at time t , λ is the coefficient of the reward term for avoiding humans, and D_t^λ is the spline maximizing the total reward given λ .

Assessing the quality of an agent’s decision depends on a space of *targets* \mathcal{Y} . Importantly, the realizations of these targets are *unknown* at the time of the decision; the agent only observes them at deployment time, after decisions are made, and in an online fashion. For example, the robot in Figure 3.1 does not know the true future state of nearby pedestrians; at any current time t , it only knows the (potentially erroneous) pedestrian predictions. In this example, \mathcal{Y} is the space of pedestrian states (e.g., 2D positions) and $y_t \in \mathcal{Y}$ is the *true* state that the pedestrian moves to at time t .

Mathematically, the quality of the decision-making is quantified by a *loss function* $\mathcal{L} : \mathcal{U} \times \mathcal{Y} \rightarrow [0, 1]$.¹ Often, the loss is more likely to be large when aggressive decisions are taken—i.e., when λ is large. Aggressive decisions can be unsafe, but taking λ too small yields conservative and under-performing decisions.

We seek an algorithm for adapting λ_t (and thus the corresponding decision D_t^λ) at each time step such that the average loss is controlled in hindsight for *any* realization of an input-target sequence $\{(x_t, y_t)\}_{t=1}^T$. This is commonly known as the *adversarial sequence model* [52, 73]. Here, our goal is to set $\lambda_{1:T}$ to achieve a *long-term risk bound*:

$$\text{find } \lambda_{1:T} \text{ s.t. } \hat{R}_T(\mathcal{D}_{1:T}, \lambda_{1:T}) \leq \varepsilon + \frac{C \cdot h(T)}{T}, \quad (3.1)$$

where ε is a pre-defined risk level in $[0, 1]$, C is a (small) constant, $h(T)$ is any sublinear function; i.e., one where $h(T)/T \rightarrow 0$ as $T \rightarrow \infty$, and

$$\hat{R}_T(\mathcal{D}_{1:T}, \lambda_{1:T}) := \frac{1}{T} \sum_{t=1}^T \mathcal{L}(D_t^{\lambda_t}(x_t), y_t) \quad \text{and} \quad \hat{R}_0 = 0.$$

We will omit $\mathcal{D}_{1:T}$ in the notation of risk when the sequence of family of decision functions is clear from the context

3.4 Theory & Conformal Controller Algorithm

In this section, we prove the core theoretical results behind Conformal Decision Theory. Specifically, we show that any sequence of families of decision functions $\mathcal{D}_{1:T}$ that are *eventually safe* can be calibrated online to achieve bounded long-term risk. We then introduce an example of a conformal controller which solves Equation (3.1) under the assumption of eventual safety.

¹The framework works for any bounded loss, but we assume the loss to be in $[0, 1]$ for simplicity.

Definition 1 (Eventually Safe). *In the setting above, we say that $\mathcal{D}_{1:T}$ is eventually safe if $\exists \varepsilon^{\text{safe}} \in [0, 1], \lambda^{\text{safe}} \in \mathbb{R}$ and a time horizon $K > 0$ such that uniformly over all sequences $\lambda_{1:K}$ and $\{(x_1, y_1), \dots, (x_k, y_k)\} \in \mathcal{X} \times \mathcal{Y}$,*

$$\begin{aligned} & \{\forall k \in [K], \lambda_k \leq \lambda^{\text{safe}}\} \\ & \implies \frac{1}{K} \sum_{k=1}^K \mathcal{L}(D_k^{\lambda_k}(x_k), y_k) \leq \varepsilon^{\text{safe}}. \end{aligned}$$

Intuitively, this condition says that there exists a safe value λ^{safe} such that if the conformal control variable lands below that value, it will incur a low risk $\varepsilon^{\text{safe}}$ after no more than K time steps. For example, even the most conservative robot planner may not be able to change its trajectory fast enough *in a single timestep*, but it could possibly do so in K time steps. For general decision-making, the existence of a safe decision function is not guaranteed, and requires domain-specific knowledge (e.g., when the loss function captures the distance between agents [17, 83, 157]). But when the decision is a *prediction set*, a safe decision function is trivial because you can always output the entire space. Note that the *eventually safe* is a strictly weaker assumption than that used for the proofs in other works, such as [5, 24, 73], which require $K = 1$. Moreover, conformal controllers are simple yet efficient algorithms that solve the Conformal Decision Theory problem stated in Equation (3.1). An example is below.

Theorem 13 (Conformal Controller). *Consider the following update rule for $\lambda_{1:T}$:*

$$\lambda_{t+1} = \lambda_t + \eta(\varepsilon - \ell_t), \forall t \in [T] \quad (3.2)$$

where $\eta > 0$ and $\ell_t := \mathcal{L}(D_t^{\lambda_t}(x_t), y_t)$.

If $\lambda_1 \geq \lambda^{\text{safe}} - \eta$ and $\mathcal{D}_{1:T}$ satisfies Definition 1 for a given $K \geq 1$ and $\varepsilon^{\text{safe}} \leq \varepsilon$, then for any realization of the data, the empirical risk is bounded:

$$\hat{R}_t(\lambda_{1:t}) \leq \varepsilon + \frac{(\lambda_1 - \lambda^{\text{safe}})/\eta + K}{t},$$

for all $t \in [K, \dots, T]$.

The update in (3.2) resembles ACI [73] and is a hybrid between the RollingRC update [66], and the P-controller update [5]. The difference is that the update is applied to λ and not the conformal quantile or quantile level.

Proof of Theorem 13. By the definition of the update rule,

$$\lambda_{t+1} = \lambda_1 + \eta \sum_{s=1}^t (\varepsilon - \ell_s).$$

By isolating $\sum_{s=1}^t \ell_s$ on one side and moving all other terms to the right-hand side, we obtain:

$$\hat{R}_t(\lambda_{1:t}) = \frac{1}{t} \sum_{s=1}^t \ell_s = \varepsilon + \frac{\lambda_1 - \lambda_{t+1}}{\eta t}.$$

To conclude, we just need to lower bound λ_t by a constant w.r.t t which is done in the following Lemma 13.1. \square

Lemma 13.1. *For the sequence in Equation 3.2, with $\lambda_1 \geq \lambda^{\text{safe}} - \eta$ we have that the parameter λ_t is bounded below by $\lambda_t \geq \lambda^{\text{safe}} - K\eta$, for all $t \in [T + 1]$.*

Proof. First note that the maximal change in the parameter is $\sup_{s \in [T]} |\lambda_{s+1} - \lambda_s| < \eta$, because $\ell_s \in [0, 1]$ and $\varepsilon \in [0, 1]$. We will then proceed by contradiction: Assume that $\inf_{s \in [T+1]} \lambda_s < \lambda^{\text{safe}} - K\eta$. Denote $t = \arg \min_{s \in [T+1]} \{\lambda_s : \lambda_s < \lambda^{\text{safe}} - K\eta\}$. That is, t is the first instant when the parameter goes below that lower bound. Then, by definition of t , $\forall s < t, \lambda_t < \lambda^{\text{safe}} - K\eta \leq \lambda_s$.

Because the max difference between successive steps is η , we can prove recursively that $\forall k \in \{0, \dots, K\}, \lambda_{t-k} < \lambda^{\text{safe}} - (K-k)\eta$. Note that, from those inequalities, we deduce that $t > K$ since $\lambda_1 \geq \lambda^{\text{safe}} - \eta$. By recursively applying the update rule $\lambda_t = \lambda_{t-K} + K\eta(\varepsilon - \frac{1}{K} \sum_{k=1}^K \ell_{t-k})$, we have:

$$\begin{aligned} & (\forall k \in \{0, \dots, K-1\}, \lambda_{t-k} < \lambda^{\text{safe}}) \\ \implies & \frac{1}{K} \sum_{k=1}^K \ell_{t-k} \leq \varepsilon^{\text{safe}} && \text{(Definition 1)} \\ \implies & \lambda_t = \lambda_{t-K} + K\eta \left(\varepsilon - \frac{1}{K} \sum_{k=1}^K \ell_{t-k} \right) \\ \implies & \lambda_t \geq \lambda_{t-K} + K\eta(\varepsilon - \varepsilon^{\text{safe}}) \\ \implies & \lambda_t \geq \lambda_{t-K}. \end{aligned}$$

Since t is the first ever timestep to go below $\lambda^{\text{safe}} - K\eta$, this is a contradiction. \square

Remark 1. *The assumption $\lambda_1 \geq \lambda^{\text{safe}} - \eta$ is not necessary to prove that $\hat{R}_t(\lambda_{1:t}) \leq \varepsilon + O(1/t)$. Intuitively, two scenarios can occur:*

1. *If $\forall k \in [K], \lambda_k \leq \lambda^{\text{safe}}$, then the empirical risk over the first K steps will be upper bounded by $\varepsilon^{\text{safe}}$. In this case, we need only to upper bound the risk between $K + 1$ and T , which can be achieved using the previous theorem or this remark.*
2. *If there exists a $k \in [K]$ such that $\lambda_k > \lambda^{\text{safe}}$, we can apply the previous theorem to upper bound the risk between k and $T + 1$. The cumulative loss between 1 and k is upper bounded by k , which is $o(1)$.*

Conformal Decision Theory in Batch

Conformal decision theory can also be applied in the so-called batch setting, wherein a separate calibration dataset is available for learning a safe decision. Here, a dataset or simulator allows for offline experimentation to quantify the risk of different decisions, e.g., offline RL. This requires a different statistical setup. Consider the case of $n + 1$ exchangeable decision functions $D_1(\lambda), \dots, D_{n+1}(\lambda)$ and an associated loss function \mathcal{L} taking a decision and returning a value in $[0, 1]$. The first n decision functions will be used for calibration of a parameter $\hat{\lambda}$ that will be used in the final decision. These exchangeable decision functions may be produced, for example, by applying a single decision function to a sequence of exchangeable data points. For the sake of simplicity, we assume that the decisions have monotone loss, i.e., that for all i ,

$$\lambda_1 \leq \lambda_2 \implies \mathcal{L}(D_i(\lambda_1)) \leq \mathcal{L}(D_i(\lambda_2)).$$

Following [9], the conformal control variable can be chosen as

$$\hat{\lambda} = \sup \left\{ \lambda : \frac{1}{n} \sum_{i=1}^n \mathcal{L}(D_i(\lambda)) \leq \epsilon - \frac{1 - \epsilon}{n} \right\}.$$

This will give a risk guarantee as a corollary of Theorem 1 of [9].

Corollary 14. *With the choice of $\hat{\lambda}$ above,*

$$\mathbb{E}[\mathcal{L}(D_{n+1}(\hat{\lambda}))] \leq \epsilon.$$

Though the validity of the algorithm follows from the theory of conformal risk control, it is substantially different in practice and deserves further study. Specifically, unlike the previous methods, in order to calculate $\hat{\lambda}$, one must iterate through a sequence of counterfactual decisions (possible values of λ) and evaluate what the loss would have been. This restricts the applications of the batch algorithm and also presents an opportunity for future work to make it more efficient and expand its scope.

3.5 Experiments

We demonstrate Conformal Decision Theory in three autonomous decision-making domains, which exhibit three different ways in which a conformal controller can be instantiated. First, we consider a robot-navigation-around-humans example in the Stanford Drone Dataset [129], where CDT tunes the robot’s reward function in an online manner to be safe but efficient. Next, we model a manufacturing setting where CDT directly calibrates the speed of the conveyor belt under a robot to achieve high-throughput and successful robot grasps. Finally, we study an automated high-frequency trading example where CDT must optimize the buying and selling of stocks.

3.5.1 Robot Navigation in Stanford Drone Dataset

Robot navigation around people must balance safety (i.e., not colliding with humans) and efficiency (i.e., the robot makes progress towards a goal). To ensure that the risk of collision is low while still making progress to the goal, the robot will calibrate its cost function at run-time using a conformal controller (CC).

Table 3.1: **Stanford Drone Dataset: Quantitative Results.** Results on the `nexus_4` scenario from SDD [129]. The robot’s goal is to cross the nexus while avoiding pedestrians. Safety was violated if the robot collided with a human. At all learning rates η , the conformal controller is more efficient at navigation than ACI in terms of time. It remains safe so long as the learning rate is set high enough so that the robot planner can quickly adapt to nearby humans; when the learning rate is set too low (near zero), proximity to humans is effectively not penalized, leading to collisions.

		Metrics									
Method	η	success	time (s)	safe	min dist (m)	avg dist (m)	5% dist (m)	10% dist (m)	25% dist (m)	50% dist (m)	
Aggressive	n/a	✓	8.567	✗	0.1595	4.058	1.253	1.546	2.495	4.021	
ACI ($\alpha = 0.01$)	0	✓	27.17	✗	0.07612	5.201	1.842	2.415	3.9	5.614	
	0.01	✓	26.67	✗	0.8026	4.575	2.261	3.014	3.507	4.574	
	0.1	✓	24.73	✗	0.7906	4.771	2.284	2.825	3.561	4.78	
Conformal Controller ($\epsilon = 2m$)	50	✓	20.03	✗	0.6122	3.299	0.8688	1.426	2.022	2.978	
	100	✓	17.4	✓	1.142	3.794	1.678	1.811	2.378	3.262	
	500	✓	17.33	✓	1.116	3.989	1.69	1.812	2.452	3.795	
	1000	✓	16.17	✓	1.265	3.599	1.698	1.81	2.282	3.303	
Conservative	n/a	✗	∞	✓	2.268	6.291	3.801	3.982	4.982	5.993	

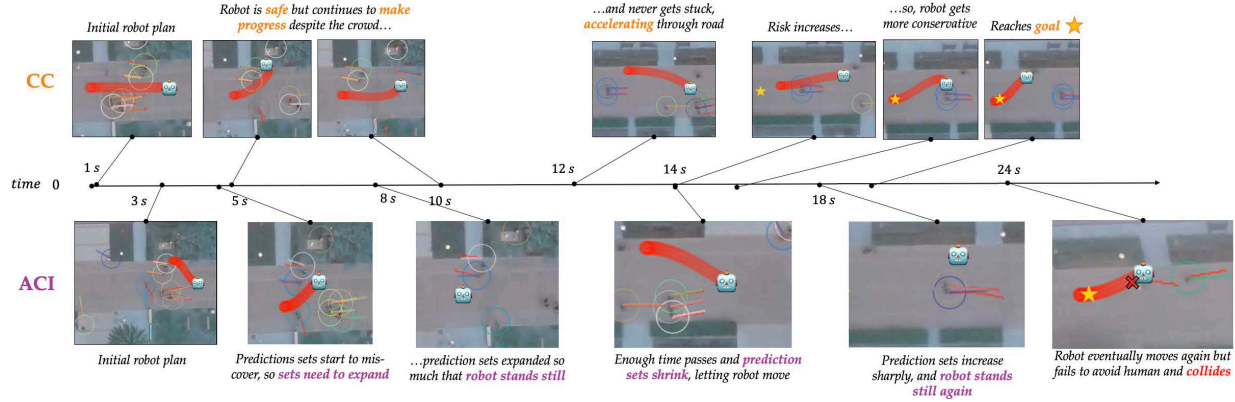


Figure 3.2: **Stanford Drone Dataset: Qualitative Results.** Visualization of interaction over time (left to right). (Top) With our conformal controller (CC), the robot always makes progress towards its goal while remaining safe, even when blocked by crowds of people. (Bottom) The ACI baseline calibrates the prediction sets. As soon as a mis-prediction happens, ACI expands the prediction sets to obtain coverage, but this frequently blocks the robot from moving anywhere (see $t = 10s$), even though the mis-predictions occurred for a pedestrian who was far away and not interfering with the robot's plan.

Decision Function & Parameterization. The robot plans via model predictive control, where at each timestep it fits a minimum-cost spline subject to its dynamic constraints, which are modeled as a nonlinear Dubins car [158]. Let $g := [g_x, g_y] \in \mathbb{R}^2$ be the robot's goal location. Let t be the current time, $H < T$ be the planning horizon, and $u_{t:t+H} \in \mathbb{R}^{H \times 3}$ be a spline consisting of the robot's planar position and orientation. The robot also gets as input the current set of short-horizon predictions of each human's state, $x_{t:t+H} \in \mathcal{P}_t$, generated by an autoregressive predictive model [85]. Note that this set \mathcal{P}_t can include predictions for *multiple* humans in the scene (as shown in Figure 3.1). The robot's planning objective is

$$J(u_{t:t+H}; \mathcal{P}_t, \lambda) := \underbrace{\sum_{\tau=t}^{t+H} \|u_{\tau}^{\text{pos}} - g\|}_{\text{Goal distance}} + \lambda \cdot \underbrace{\left(- \inf_{x_{\tau} \in \mathcal{P}_t} \|u_{\tau}^{\text{pos}} - x_{\tau}\|\right)}_{\text{Human avoidance}},$$

where the notation $u_{\tau}^{\text{pos}} \in \mathbb{R}^2$ indicates the xy -positional entries of the robot's state at time τ . Note that the conformal control variable λ scales the cost of staying far away from predicted human states: if $\lambda = 0$ the robot only cares about reaching the goal; if $\lambda > 0$ then the robot is increasingly penalized for intersecting with predicted human trajectories. The decision function outputs the minimum-cost trajectory for the robot

$$D_t^{\lambda} := \arg \min_{u_{t:t+H} \in \mathcal{U}} J(u_{t:t+H}; \mathcal{P}_t, \lambda),$$

where \mathcal{U} is the set of feasible splines (ones that are dynamically feasible for the robot and also do not intersect with environment obstacles). At the next timestep, the robot re-predicts the human trajectory (i.e., generates \mathcal{P}_{t+1}) and re-plans the decision D_{t+1}^{λ} .

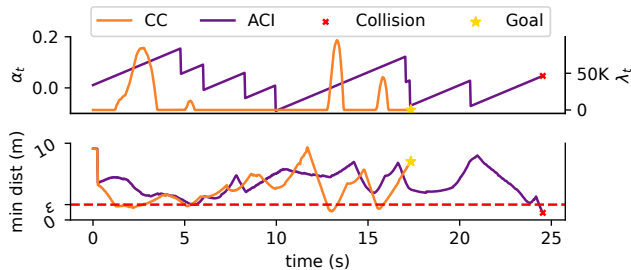


Figure 3.3: **Stanford Drone Dataset.** (Top) Trajectories of λ_t (calibrated by **CC**) and α_t (from **ACI** to calibrate sets). When $\alpha_t \leq 0$, **ACI** returns infinite set and the robot stops. (Bottom) Distance to the nearest human over time. λ_t is large when the robot is close to human, while α_t is unrelated. The λ_t trajectory is shorter because it reaches the goal faster.

Loss Function. Let $\mathcal{Y} \subset \mathbb{R}^2$ and the targets $y_t^1, \dots, y_t^M \in \mathcal{Y}$ be the actual xy positions of each of the M humans that the robot observes at time t . The loss function is defined as the negative distance to the nearest human,

$$\mathcal{L} := - \inf_{i \in [M]} \|y_t^i - u_t^{\text{pos}}\|_2,$$

where u_t^{pos} is the robot’s current position. To make this value bounded, we clip the loss to the size of the video. Note that because we use a negative loss, we also changed λ so that the larger λ , the more conservative the decision.

Metrics. We measure a boolean *safe* variable indicating if the robot did not ever collide with a human. We also measure a boolean *success* variable if the robot reached the goal location by the end of the interaction episode (i.e., length of video in the dataset). We also measure the time to reach the goal location and the minimum, mean, and $\{5\%, 10\%, 25\%, 50\%\}$ quantiles of the distance to the nearest human.

Experimental Setup. All methods are evaluated on interactions from the `nexus_4` video in the Stanford Drone Dataset (SDD) [129]. The risk threshold is $\varepsilon = 2m$ (i.e., radius around human). The robot always starts from the same initial condition and moves to the same goal. This scenario has a high density of pedestrians, making the risk-performance tradeoff for the robot nontrivial. Our approach (**CC**) adapts the reward weight λ_t on the human collision cost based on Equation 3.2 so that the decision risk is calibrated. Our baseline robot planners: **conservative** which always uses the safe decision function $D_t^{\lambda=1}$, **aggressive** which uses $D_t^{\lambda=0}$, and **ACI** [58] which first uses adaptive conformal prediction to calibrate prediction sets and then plans to avoid these sets.

Results. Quantitative results shown in Table 3.1 and qualitative results in Figure 3.1. Because the **conformal controller** calibrates the robot’s decisions directly, it is substantially ($\sim 29\%$) faster at reaching the goal than the **ACI** algorithm (see visualization over time in Figure 3.2). While the **aggressive** baseline reaches the goal fastest, it consistently violates the safety threshold. On the other hand, the **conservative** baseline never completes the

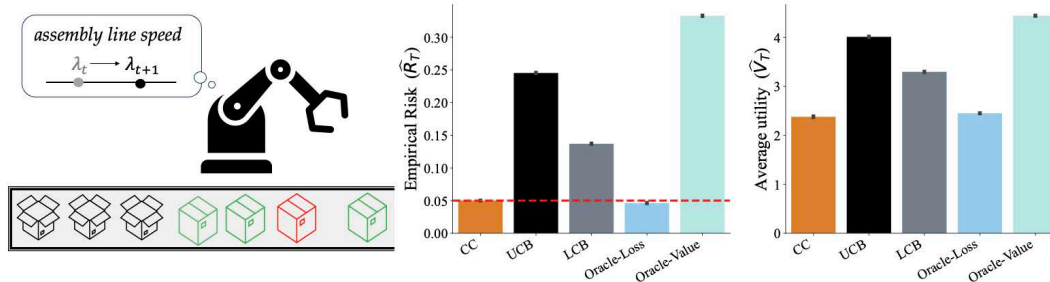


Figure 3.4: **Manufacturing Assembly Line Robot: Quantitative Results.** (Left) Illustrative example: Robot must adjust the speed so that it grasps the most items while minimizing grasp failure. (Right) Empirical risk \hat{R}_T , and average utility (i.e., successful grasps), \hat{V}_T on 1000 runs. Our method is denoted by (CC). Dashed red line is target risk $\varepsilon = 0.05$.

task, getting stuck far away from the crowds of pedestrians. The **conformal controller** ensures safety so long as the learning rate is fast enough for the robot planner to quickly adapt to changes in nearby human behavior (see Figure 3.3). Note that **ACI** can result in collisions for two reasons: 1) the prediction sets do not adapt fast enough for the spline planner to react and swerve out of the way of the pedestrian, 2) if the prediction sets become so large that there is no feasible spline and the robot must stand in place, the pedestrians sometimes run into the robot. This issue was independently observed in [58].

3.5.2 Manufacturing Assembly Line Robot

Consider a factory assembly line where a robot has to grab items from a conveyor belt (left, Figure 3.4). As the speed increases, the throughput of items increases but so does the ratio of robot grasp failures. The agent must calibrate the speed so that the ratio of failures over time stays below ε .

Decision Function & Parameterization. The agent directly modifies the speed, thus the action is defined as $u_t := \lambda_t$. Here we take $\lambda_t \in [0, 1]$.

Risk Function. For a given conveyor belt speed λ , the robot will attempt to grab $n(\lambda)$ items, among which $d(\lambda)$ are failed grasps. The loss received by the robot will be $\mathcal{L}(\lambda) := d(\lambda)/n(\lambda)$.

Metrics. We measure average utility (i.e., # of successful grasps), $\hat{V}_T := \frac{1}{T} \sum_{t=1}^T V(\lambda_t)$, and empirical risk, $\hat{R}_T(\lambda_{1:T})$.

Experimental Setup. We assume that the number of items $n(\lambda)$ the robot attempts to grab is drawn as $\text{Pois}(C \cdot \sqrt{\lambda})$. The number of failed grasps conditioned on the total number of items is $d(\lambda)|n \sim \text{Bin}(n, C' \cdot \lambda)$. Importantly, the distributions of n, d , and the parameters C, C' are all *unknown to the agent*. Our conformal controller method (**CC**) adjusts the speed

λ_t based on the update rule from Equation 3.2. In addition to the risk function, we also track a utility function which is the number of successful grasps $V(\lambda) := n(\lambda) - d(\lambda)$.

We compare our method with two baselines: A bandit algorithm running the upper confidence bound algorithm (**UCB**) [98] to maximize the utility V and another algorithm running the lower confidence bound algorithm (**LCB**) to minimize the loss \mathcal{L} . We also add two methods with oracle access to the otherwise unknown parameters: **Oracle-Value** selects the best speed to maximize grasp success $\lambda_V^* := \arg \max_{\lambda} \mathbb{E}[V(\lambda)]$ and **Oracle-Loss** selects the best speed $\lambda_{\mathcal{L}}^*$ such that $\mathbb{E}[\mathcal{L}(\lambda_{\mathcal{L}}^*)] := \varepsilon$. The values selected for the parameters are in Figure 3.4. We run all methods for a horizon $T = 2000$, set $C = 10$, $C' = 0.2$, and the target risk is $\varepsilon = 0.05$ (i.e., $\leq 5\%$ failed grasp).

Results. We run the simulation $N = 1000$ times, and calculate the average empirical risk and the average number of successful grasps. In Figure 3.4, we find that our method performs as well as the **Oracle-Loss**, ensuring that the empirical risk of grasps never exceeds $\varepsilon = 0.05$, while still ensuring high throughput of successfully grasped items. **UCB** and **LCB** both violate the empirical risk threshold: **UCB** incurs this risk but achieves a higher number of successful grasps, while **LCB** is slow to learn its target, resulting in a higher risk over the time horizon.

3.5.3 Stock Trading Agent

We consider an automated trading agent that trades a stock at high frequency. We model the agent as able to either buy or short-sell the stock, with no trading cost. When buying the stock at time t , the agent receives return r_t . When short-selling the stock, the agent receives a return $-r_t$. The agent must calibrate its trading decisions so the annualized loss is at or beneath the investor’s loss threshold of $\varepsilon\%$.

Decision Function & Parameterization. At every timestep, t , the agent has access to the past history of returns and its own actions. The agent can use it to construct a confidence set \hat{C}_{λ} where λ is the conformal control variable. Given a predicted set, the agent can decide to either buy if the entire set is above zero, short-sell if the entire set is below zero, and not do anything if zero is in the set:

$$D_t^{\lambda} := \begin{cases} 1 & \text{if } \min(\hat{C}_{\lambda}) > 0 \\ -1 & \text{if } \max(\hat{C}_{\lambda}) < 0 \\ 0 & \text{o.w.} \end{cases}$$

Risk Function. The agent’s action is $u \in \{-1, 0, 1\}$ which incurs a loss $\mathcal{L}(u, r) := -u \cdot r \cdot 1\{u \cdot r < 0\}$, i.e., the agent suffers a loss equal to the amount of money lost by that decision. We clip the loss to make it bounded.

Experimental Setup. We simulate stock returns using a geometric Brownian motion. We assume that we observe returns every hour, so we have $n = 252(\text{days}) \times 7(\text{hours per day})$

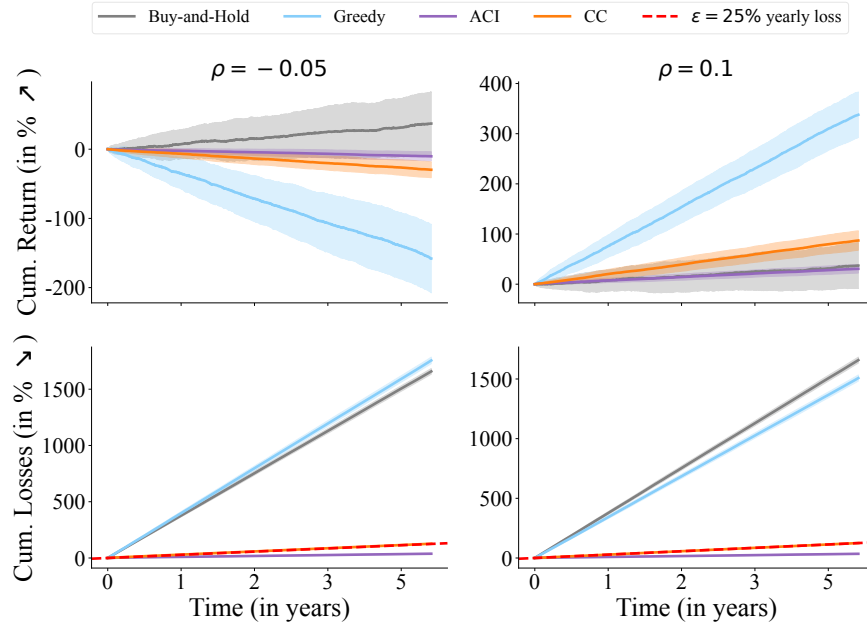


Figure 3.5: **Stock Trading: Quantitative Results.** All results over 5 year period. The yearly loss threshold $\varepsilon = 25\%$. (left) Despite a poor prediction model of return (negative correlation), the CC achieves bounded loss at the user’s threshold (bottom, dashed red line overlaps with orange CC line) but is not the best at keeping the return the highest. (right) With a strong prediction model on the return (positive correlation), the CC is able to achieve high yearly returns (second only to Greedy) while simultaneously respecting the loss threshold (which the Greedy violates).

steps per year:

$$r_t := \mu\Delta + \sigma\sqrt{\Delta}Z_t \quad \text{where } \Delta = 1/n.$$

We assume that at time $t-1$, the agent has access to a prediction \hat{r}_t and we assume that the correlation $\text{corr}(r_t, \hat{r}_t) := \rho$. The higher ρ the better the predicted returns \hat{r}_t . The predicted interval is

$$\hat{C}_\lambda(\hat{r}_t) := [\hat{r}_t - \sigma\sqrt{\Delta}z_{\lambda/2}, \hat{r}_t + \sigma\sqrt{\Delta}z_{1-\lambda/2}],$$

where z_λ is the quantile of level λ of the normal distribution.

Metrics. In addition to the loss, we also measure return $V(u, r) := u \cdot r$ when the agent’s action is u .

Results. We run $N = 100$ simulations over five years. We set $\mu = 0.08, \sigma = 0.2$, which are approximately the historical values for the S&P 500. We compare our **CC** method with: the **Buy-and-Hold** strategy that simply buys the stock at each timestep, the **Greedy** strategy that buys the stock whenever the prediction is above zero and short-sells it when the prediction is below zero (equivalent to $D(\lambda = 1)$), and **ACI** that adjusts λ online using

the ACI algorithm. We set the target coverage for ACI at 90% and our annualized loss threshold to be less than $\varepsilon = 25\%$ (the threshold per time-step is therefore ε/n). For the prediction of returns, we simulate another geometric Brownian motion,

$$\hat{r}_t := \mu\Delta + \sigma\sqrt{\Delta}W_t \quad \text{where} \quad \text{corr}(W_t, Z_t) = \rho.$$

The results for the different methods are in Figure 3.5. We plot the cumulative return and cumulative loss for all methods and for two models: $\rho = 0.1$ (good model) and $\rho = -0.05$ (bad model). In both cases, our **CC** quickly adapts the parameter to stay below the loss threshold, while having good returns when the predictive model is good ($\rho = 0.1$). The **Greedy** approach has more extreme returns (negative when the model is bad, positive when the model is good) with a high level of loss. **ACI** is highly conservative, resulting in smaller loss, significantly below the threshold. By being so conservative, the algorithm limits its potential gain when the predictive model is actually good. **Buy-and-hold** also has high cumulative loss as it moves with the stock, and has a more consistent return, as it is independent of the model quality.

3.6 Discussion & Conclusion

In this chapter, we introduce *Conformal Decision Theory*, a theoretical and algorithmic framework for producing safe decisions despite being based on imperfect machine-learning predictions. We have described our method in both the online adversarial setting, and also the batch exchangeable setting. The main difference between the two is that the online algorithms we present are computationally trivial, while the batch setting can require evaluating a large amount of *counterfactual* decisions (indexed by different choices of λ) on every calibration point. Though this can be done with binary search, it still presents operational challenges. One path for future work may be to test the method in settings where simulators or data sets can support this form of offline policy evaluation. Another may be to develop formally valid approximations of the batch technique which preserve risk control while being more practical. Furthermore, extensions of the batch technique to non-exchangeable settings are readily available, e.g., by use of the techniques in [65], and could be evaluated.

Finally, despite λ being 1-dimensional, our procedure can index an arbitrary set of decisions. Consider a set of decisions \mathcal{D} , a utility predictor $\hat{u}(d; x)$ where $d \in \mathcal{D}$, and a loss predictor $\hat{\mathcal{L}}(d; x)$, we can maximize utility subject to the constraint that our predicted loss is controlled:

$$\begin{aligned} D_t &= \arg \max_{d \in \mathcal{D}} \hat{u}(d; x_t) \\ &\text{s.t.} \quad \hat{\mathcal{L}}(d; x_t) \leq \lambda_t. \end{aligned}$$

This will work as long as we revert to a safe decision if $\lambda_t \leq \lambda^{\text{safe}}$; where the sequence λ_t is defined in Equation 3.2. However, no guarantees on utility are provided. This topic would be a great avenue for future work, bringing conformal prediction closer to the classical statistical decision theory of Lehmann [99], von Neumann and Morgenstern [152], and others.

Part II

Prediction-Powered Inference

Chapter 4

Core Methodology

4.1 Introduction

Imagine a scientist has a machine-learning system that can supply accurate predictions about a phenomenon far more cheaply than any gold-standard experimental technique. The scientist may wish to use these predictions as evidence in drawing scientific conclusions. For example, accurate predictions of three-dimensional structures have been made for a vast catalog of known protein sequences [89, 148] and are now being used in proteomics studies [20, 25]. Such machine-learning systems are increasingly common in modern scientific inquiry, in domains ranging from cancer prognosis to microclimate modeling. Predictions are not perfect, however, which may lead to incorrect conclusions. Moreover, as predictions beget other predictions, these imperfections may cumulatively amplify. How can modern science leverage machine-learning predictions in a statistically principled way?

One way to use predictions is to follow the imputation approach: proceed as if they are gold-standard measurements. Although this lets the scientist draw conclusions cheaply and quickly due to the high-throughput nature of the machine-learning system, the conclusions may be invalid because the predictions may have biases.

Another approach is to apply the classical approach: ignore the machine-learning predictions and only use the available gold-standard measurements, which are typically far less abundant than predictions. The resulting discoveries will be statistically valid, but the smaller amount of data will limit the scope of possible discoveries.

This chapter presents *prediction-powered inference*, a framework that achieves the best of both worlds: extracting information from the predictions of a high-throughput machine-learning system, while guaranteeing statistical validity of the resulting conclusions. Prediction-powered inference provides a protocol for combining predictions, which are abundant but not always trustworthy, with gold-standard data, which is trusted but scarce, to compute confidence intervals and p-values. The resulting confidence intervals and p-values are statistically valid, as in the classical approach, but also leverage the information contained in the pre-

dictions, as in the imputation approach, to make the confidence intervals smaller and the p-values more powerful.

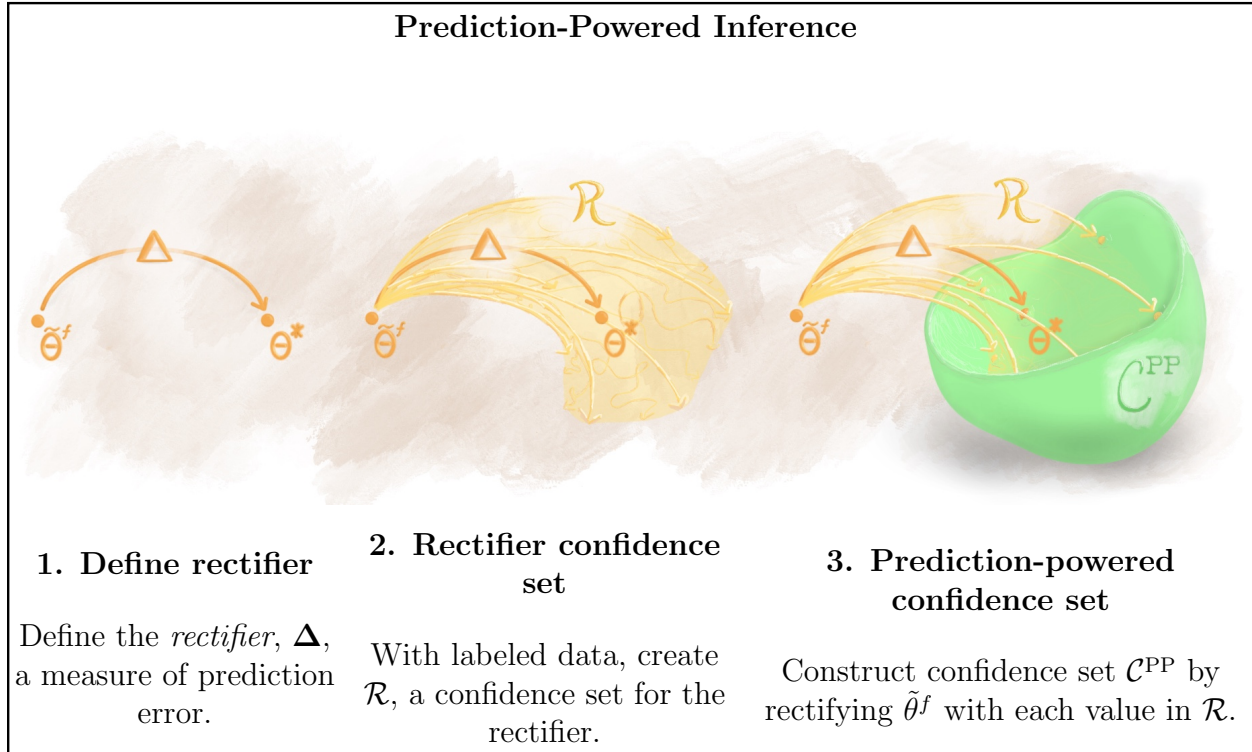
Prediction-powered inference can be used with any machine-learning system. As such, it absolves the need for case-by-case analyses dependent on the machine-learning algorithm on hand. The proposed protocol thereby enables researchers to report and assess the evidence for their conclusions in a fully standardized way.

4.1.1 General principle

We now overview prediction-powered inference. The goal is to estimate a quantity θ^* , such as the mean or median value of a random outcome over a population of interest. Towards this goal, we have access to a small gold-standard dataset of paired features and outcomes, $(X, Y) = ((X_1, Y_1), \dots, (X_n, Y_n))$, as well as the features from a large unlabeled dataset, $(\tilde{X}, \tilde{Y}) = ((\tilde{X}_1, \tilde{Y}_1), \dots, (\tilde{X}_N, \tilde{Y}_N))$, where we do not observe the true outcomes $\tilde{Y}_1, \dots, \tilde{Y}_N$. We care about the case where $N \gg n$. For both datasets, we have predictions of the outcome made by a machine-learning algorithm f , denoted $f(X) = (f(X_1), \dots, f(X_n))$ and $f(\tilde{X}) = (f(\tilde{X}_1), \dots, f(\tilde{X}_N))$.

Prediction-powered inference builds confidence intervals that are guaranteed to contain θ^* . Imagine we have an estimator $\hat{\theta}$ of θ^* . One feasible but naive way to estimate θ^* , which we call the imputation approach, is to treat the predictions as gold-standard outcomes and compute $\tilde{\theta}^f = \hat{\theta}(\tilde{X}, f(\tilde{X}))$. If the predictions are accurate, meaning $f(\tilde{X}_i) \approx \tilde{Y}_i$, then $\tilde{\theta}^f$ is close to θ^* . However, $\tilde{\theta}^f$ will generally be biased due to errors in the predictions. Instead, our key idea is to use the gold-standard dataset to quantify how the prediction errors affect the imputed estimate, and then construct a confidence set for θ^* by adjusting for this effect.

More systematically, the first step is to introduce a problem-specific measure of prediction error called the *rectifier*, denoted as Δ . The rectifier captures how errors in the predictions lead to bias in $\tilde{\theta}^f$. Intuitively, Δ recovers θ^* by “rectifying” $\tilde{\theta}^f$. The appropriate rectifier depends on the estimand of interest θ^* , and we show how to derive it for a broad class of estimands. Next, we use the gold-standard data to construct a confidence set for the rectifier, \mathcal{R} . Finally, we form a confidence set for θ^* by taking $\tilde{\theta}^f$ and rectifying it with each possible value in the set \mathcal{R} . The collection of these rectified values is the prediction-powered confidence set, \mathcal{C}^{PP} , which is guaranteed to contain θ^* with high probability.



Prediction-powered inference leads to powerful and provably valid confidence intervals and p-values for a broad class of statistical problems, enabling researchers to reliably incorporate machine learning into their analyses. We provide practical algorithms for constructing prediction-powered confidence intervals for means, quantiles, modes, linear and logistic regression coefficients, as well as other inferential targets. For conciseness, our technical statements and algorithms will focus on constructing confidence intervals; however, note that through the duality between confidence intervals and hypothesis tests, our intervals directly imply valid prediction-powered p-values and hypothesis tests as well.

4.1.2 Further preliminaries

We use $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^n$ to denote the labeled dataset, where $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_n)$. We use the terms “labeled” and “gold-standard” interchangeably. We use analogous notation for the unlabeled dataset, $(\tilde{X}, \tilde{Y}) \in (\mathcal{X} \times \mathcal{Y})^N$, where the outcomes \tilde{Y} are not observed. For now we assume that (X, Y) and (\tilde{X}, \tilde{Y}) are independently and identically distributed samples from a common distribution, \mathbb{P} . We generalize our results to settings with distribution shift and finite populations in Section 4.4.2 and Appendix 4.5.2, respectively. By θ^* we denote the estimand of interest, which will typically be an underlying property of \mathbb{P} , such as the mean outcome.

Next, we have a prediction rule, $f : \mathcal{X} \rightarrow \mathcal{Y}$, that is independent of the observed data. For example, it may have been trained on other data independent from both the labeled and the unlabeled data. Thus, $f(X_i)$ denote the predictions for the labeled data and $f(\tilde{X}_i)$ denote the predictions for the unlabeled data. We let $f(X) = (f(X_1), \dots, f(X_n))$ and $f(\tilde{X}) = (f(\tilde{X}_1), \dots, f(\tilde{X}_N))$. We will treat $X, Y, \tilde{X}, \tilde{Y}, f(X), f(\tilde{X})$ as vectors and matrices where appropriate.

Our key conceptual innovation is the *rectifier* Δ : a measure of the prediction rule's error. We formally define the rectifier in Section 4.2. We use $\hat{\Delta}$ to denote an estimate of the rectifier based on labeled data, which we call the empirical rectifier.

4.1.3 Warmup: Mean estimation

Before presenting our main results, we use the example of mean estimation to build intuition. Our goal is to give a valid confidence interval for the average outcome, $\theta^* = \mathbb{E}[Y_i]$. The classical estimate of θ^* is the sample average of the outcomes on the labeled dataset, $\hat{\theta}^{\text{class}} = \frac{1}{n} \sum_{i=1}^n Y_i$. We construct a prediction-powered estimate, $\hat{\theta}^{\text{PP}}$, and show that it leads to tighter confidence intervals than $\hat{\theta}^{\text{class}}$ if the prediction rule is accurate. Consider

$$\hat{\theta}^{\text{PP}} = \underbrace{\frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i)}_{\tilde{\theta}^f} - \underbrace{\frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)}_{\hat{\Delta}}.$$

The key idea is that if the predictions are accurate, we have $\hat{\Delta} \approx 0$ and $\hat{\theta}^{\text{PP}} \approx \frac{1}{N} \sum_{i=1}^N \tilde{Y}_i$, which has a much lower variance than $\hat{\theta}^{\text{class}}$ since $N \gg n$.

Notice $\hat{\theta}^{\text{PP}}$ is unbiased for θ^* and it is a sum of two independent terms. Thus, we can construct 95% confidence intervals for θ^* as

$$\underbrace{\hat{\theta}^{\text{PP}} \pm 1.96 \sqrt{\frac{\hat{\sigma}_{f-Y}^2}{n} + \frac{\hat{\sigma}_f^2}{N}}}_{\text{prediction-powered interval}} \quad \text{or} \quad \underbrace{\hat{\theta}^{\text{class}} \pm 1.96 \sqrt{\frac{\hat{\sigma}_Y^2}{n}}}_{\text{classical interval}},$$

where $\hat{\sigma}_Y^2$, $\hat{\sigma}_{f-Y}^2$, and $\hat{\sigma}_f^2$ are the estimated variances of the Y_i , $f(X_i) - Y_i$, and $f(\tilde{X}_i)$, respectively. The prediction-powered confidence interval is smaller than the classical interval when the model is good. Because $N \gg n$, the width of the prediction-powered interval is primarily determined by the term $\hat{\sigma}_{f-Y}^2$. Furthermore, when the model has small errors, we have $\hat{\sigma}_{f-Y}^2 \ll \hat{\sigma}_Y^2$. Thus, the width of the prediction-powered interval will be smaller than the width of the classical interval. This estimator exists in many forms in the literature—see Section 4.1.4. This variance reduction is why prediction-powered confidence intervals are smaller than their classical counterparts in a broad range of settings beyond mean estimation.

4.1.4 Related work

Our technical results generalize tools from the model-assisted survey sampling literature [e.g., 137], which provides methods to improve inference from surveys in the presence of auxiliary information. In particular, the mean estimator in Section 4.1.3 is the difference estimator, closely related to generalized regression estimators [38]. It has long been recognized that model predictions can be leveraged as auxiliary data [166], and much work has gone into producing asymptotically valid confidence intervals when the predictive model is fit on the same data that is used for inference—see [28] for a recent overview. Our work is also related to the statistical literature on semiparametric inference, missing data, and multiple imputation [e.g., 107]. In particular, Robins et al. [131], Robins and Rotnitzky [130], Chen and Breslow [44], Yu and Nan [171] study regression with missing data. The rectifier resembles debiasing strategies that are pervasive in this literature, an example being the AIPW estimator [130]. Likewise, our setting is related to measurement error [e.g., 37], particularly to Chen et al. [47], who study the estimation of parameters defined as solutions to many estimating equations, as we will in this chapter. Prediction-powered inference aims to provide simple, broadly applicable algorithms using similar debiasing tricks, while allowing the use of state-of-the-art black-box machine-learning systems.

Recently, a body of work on estimation with many unlabeled data points and few labeled data points has been developed [16, 45, 124, 144, 160, 174], focusing on efficiency in semiparametric or high-dimensional regimes. In particular, Chakraborty and Cai [40] study efficient estimation of linear regression parameters, Chakraborty et al. [41, 42] study efficient quantile estimation and quantile treatment effect estimation with high-dimensional covariates, Zhang and Bradic [175] study mean estimation in a high-dimensional setting, Deng et al. [56] study linear regression parameters in a high-dimensional setting, and Hou et al. [82] study an imputation approach to improving generalized linear models. Finally, Song et al. [144] study M-estimation, using a projection-based correction to the classical M-estimator loss based on simple statistics (e.g. low-order polynomials) of the features. Prediction-powered inference continues in this vein but focuses on the setting where the scientist has access to a good predictive model fit on separate data and makes no assumptions about the model (such as consistency). The confidence intervals and resulting p-values from previous work rely on asymptotic approximations, while prediction-powered inference has both asymptotic and nonasymptotic variants. Furthermore, prediction-powered inference goes beyond random sampling and considers certain forms of distribution shift.

More distantly, our setting, in which we have access to some labeled data alongside unlabeled data, also appears in semisupervised learning [e.g., 177, 178], which studies the question of how to improve prediction accuracy with unlabeled data. We also refer the reader to the related literatures on transfer learning [e.g., 21, 103, 145, 167] and surrogates in causal inference [e.g., 90]. Thematically, our work is most similar to the work of Wang et al. [159], who also introduce a method to correct machine-learning predictions for the purpose of subsequent inference. However, our work provides confidence intervals that are provably

valid under minimal assumptions about the data-generating distribution, whereas Wang et al. require certain parametric assumptions about the relationship between the prediction model and the true response. We compare against this baseline in Appendix 4.5.24.

4.2 Main theory: Convex estimation

Our main contribution is a technique for inference on estimands that can be expressed as the solution to a *convex optimization problem*. In addition to means, this includes medians, other quantiles, linear and logistic regression coefficients, and many other quantities. Formally, we consider estimands of the form

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^p} \mathbb{E}[\ell_\theta(X_i, Y_i)],$$

for a loss function $\ell_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that is convex in $\theta \in \mathbb{R}^p$, for some $p \in \mathbb{N}$. Throughout, we take the existence of θ^* as given. If the minimizer is not unique, our method will return a confidence set guaranteed to contain all minimizers. Under mild conditions, convexity ensures that θ^* can also be expressed as the value solving

$$\mathbb{E}[g_{\theta^*}(X_i, Y_i)] = 0, \tag{4.1}$$

where $g_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^p$ is a subgradient of ℓ_θ with respect to θ . We will call convex estimation problems where θ^* satisfies (4.1) nondegenerate, and we will later discuss mild conditions that ensure this regularity.

Defining the rectifier. Following the outline in Section 4.1.1, the first step in prediction-powered inference is to define a rectifier. As in the mean estimation case, the rectifier captures a notion of prediction error. In the general setting of convex estimation problems, the relevant notion of error is the bias of the subgradient g_θ computed using the predictions:

$$\Delta_\theta = \mathbb{E}[g_\theta(X_i, Y_i) - g_\theta(X_i, f(X_i))]. \tag{4.2}$$

Rectifier confidence set. The second step is to create a confidence set for the rectifier, $\mathcal{R}_\delta(\theta)$, satisfying

$$P(\Delta_\theta \in \mathcal{R}_\delta(\theta)) \geq 1 - \delta.$$

Because the rectifier is an expectation for each θ , $\mathcal{R}_\delta(\theta)$ can be constructed using standard, off-the-shelf confidence intervals for the mean, which we review in Appendix 4.5.21.

Prediction-powered confidence set. The final step is to form a confidence set for θ^* . We do so by combining $\mathcal{R}_\delta(\theta)$ with a term that accounts for finite-sample fluctuations due to having N unlabeled data points. In particular, for every θ , we want a confidence set $\mathcal{T}_{\alpha-\delta}(\theta)$ for $g_\theta^f = \mathbb{E}[g_\theta(X_i, f(X_i))]$, satisfying

$$P(g_\theta^f \in \mathcal{T}_{\alpha-\delta}(\theta)) \geq 1 - (\alpha - \delta).$$

Again, since g_θ^f is a mean, constructing $\mathcal{T}_{\alpha-\delta}(\theta)$ is easy and can be done with off-the-shelf tools.

We put all the steps together in Theorem 15.

Theorem 15 (Convex estimation). *Suppose that the convex estimation problem is nondegenerate as in (4.1). Fix $\alpha \in (0, 1)$ and $\delta \in (0, \alpha)$. Suppose that, for any $\theta \in \mathbb{R}^p$, we can construct $\mathcal{R}_\delta(\theta)$ and $\mathcal{T}_{\alpha-\delta}(\theta)$ satisfying*

$$P(\Delta_\theta \in \mathcal{R}_\delta(\theta)) \geq 1 - \delta; \quad P(g_\theta^f \in \mathcal{T}_{\alpha-\delta}(\theta)) \geq 1 - (\alpha - \delta).$$

Let $\mathcal{C}_\alpha^{\text{PP}} = \{\theta : 0 \in \mathcal{R}_\delta(\theta) + \mathcal{T}_{\alpha-\delta}(\theta)\}$, where $+$ denotes the Minkowski sum.¹ Then,

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

This result means that we can construct a valid confidence set for θ^* , without assumptions about the data distribution or the machine-learning model, for any nondegenerate convex estimation problem. We also present an asymptotic counterpart of Theorem 15 in Appendix 4.5.6.

Most practical problems are nondegenerate (4.1). For example, if the loss is differentiable for all $\theta \in \mathbb{R}^p$, then the problem is immediately nondegenerate. Furthermore, if the data distribution does not have point masses and, for every θ , $\ell_\theta(x, y)$ is nondifferentiable only for a measure-zero set of (x, y) pairs, then the problem is again nondegenerate.

We have focused on convex estimation problems, since this is a broad class of estimands addressed by prediction-powered inference. Nonetheless, we highlight that the general principles for prediction-powered inference from Section 4.1.1 are applicable more broadly, and lead to additional results and algorithms for other estimands and some forms of distribution shift; see Section 4.4 for such extensions.

4.2.1 Algorithms

In this section we present prediction-powered algorithms for several canonical inference problems. We defer the proofs of their validity to Appendix 4.5.9. The algorithms rely on confidence intervals derived from the central limit theorem. We implicitly assume the standard, mild regularity conditions required for the asymptotic validity of such intervals, which we overview in Appendix 4.5.8. We also present a parallel set of algorithms that are obtained via nonasymptotic constructions in Appendix 4.5.7. In the algorithms we use $z_{1-\delta}$ to denote the $1 - \delta$ quantile of the standard normal distribution, for $\delta \in (0, 1)$. All algorithms are technically simplified versions of Algorithm 5 with different choices of gradients and rectifiers; see Table 4.1 for the correspondence.

¹The Minkowski sum of two sets A and B is equal to $\{a + b : a \in A, b \in B\}$.

Mean estimation. We begin by returning to the problem of mean estimation:

$$\theta^* = \mathbb{E}[Y_i]. \quad (4.3)$$

The mean can alternatively be expressed as the solution to a convex optimization problem by writing it as the minimizer of the average squared loss:

$$\theta^* = \arg \min_{\theta \in \mathbb{R}} \mathbb{E}[\ell_\theta(Y_i)] = \arg \min_{\theta \in \mathbb{R}} \mathbb{E} \left[\frac{1}{2} (Y_i - \theta)^2 \right].$$

The squared loss $\ell_\theta(y)$ is differentiable, with gradient equal to $g_\theta(y) = \theta - y$. Applying this in the definition of the rectifier (4.2), we get $\Delta_\theta \equiv \Delta = \mathbb{E}[f(X_i) - Y_i]$. Note that this rectifier has no dependence on θ . We provide an explicit algorithm for prediction-powered mean estimation and its guarantee in Algorithm 1 and Proposition 16, respectively.

Proposition 16 (Mean estimation). *Let θ^* be the mean outcome (4.3). Then, the prediction-powered confidence interval in Algorithm 1 has valid coverage: $\liminf_{n, N \rightarrow \infty} P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha$.*

Quantile estimation. We now turn to quantile estimation. For a pre-specified level $q \in (0, 1)$, we wish to estimate the q -quantile of the outcome distribution:

$$\theta^* = \min \{ \theta : P(Y_i \leq \theta) \geq q \}. \quad (4.4)$$

To simplify the exposition, we assume that the distribution of Y_i does not have point masses; this ensures that the problem is nondegenerate (4.1), though it is possible to generalize beyond this setting with a standard construction. It is well known [93] that the q -quantile can be expressed in variational form as

$$\theta^* = \arg \min_{\theta \in \mathbb{R}} \mathbb{E}[\ell_\theta(Y_i)] = \arg \min_{\theta \in \mathbb{R}} \mathbb{E} [q(Y_i - \theta) \mathbb{1}\{Y_i > \theta\} + (1 - q)(\theta - Y_i) \mathbb{1}\{Y_i \leq \theta\}],$$

where ℓ_θ is called the quantile loss (or “pinball” loss). The quantile loss has subgradient $g_\theta(y) = -q \mathbb{1}\{y > \theta\} + (1 - q) \mathbb{1}\{y \leq \theta\} = -q + \mathbb{1}\{y \leq \theta\}$. Plugging the expression for $g_\theta(y)$ into the definition (4.2), we get the relevant rectifier: $\Delta_\theta = P(Y_i \leq \theta) - P(f(X_i) \leq \theta) = \mathbb{E}[\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\}]$. In Algorithm 2 we state an algorithm for prediction-powered quantile estimation; see Proposition 17 for a statement of validity.

Proposition 17 (Quantile estimation). *Let θ^* be the q -quantile (4.4). Then, the prediction-powered confidence set in Algorithm 2 has valid coverage: $\liminf_{n, N \rightarrow \infty} P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha$.*

Logistic regression. In logistic regression, the target of inference is defined by

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \mathbb{E}[\ell_\theta(X_i, Y_i)] = \arg \min_{\theta \in \mathbb{R}^d} \mathbb{E}[-Y_i \theta^\top X_i + \log(1 + \exp(\theta^\top X_i))], \quad (4.5)$$

where $Y_i \in \{0, 1\}$. The logistic loss is differentiable and hence the optimality condition (4.1) is ensured. Its gradient is equal to $g_\theta(x, y) = -xy + x\mu_\theta(x)$, where $\mu_\theta(x) = 1/(1 + \exp(-x^\top\theta))$ is the predicted mean for point $x \in \mathcal{X}$ based on parameter vector θ . Other generalized linear models (GLMs) have the same gradient form, and thus also optimality condition (4.1), but for a different mean predictor $\mu_\theta(x)$ (see Chapter 3 of Efron [61]). For example, Poisson regression uses $\mu_\theta(x) = \exp(x^\top\theta)$. In view of our general solution for convex estimation, the rectifier is constant for all θ and equal to $\Delta_\theta \equiv \Delta = \mathbb{E}[X_i(f(X_i) - Y_i)]$. In Algorithm 3 we state a method for prediction-powered logistic regression and in Proposition 18 we provide its guarantee. We use $X_{i,j}$ to denote the j -th coordinate of point X_i . Poisson regression is handled in essentially the same way: concretely, in Algorithm 3 we simply change the choice of $\mu_\theta(x)$ defined in line 5.

Proposition 18 (Logistic regression). *Let θ^* be the logistic regression solution (4.5). Then, the prediction-powered confidence set in Algorithm 3 has valid coverage: $\liminf_{n,N \rightarrow \infty} P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha$.*

Linear regression. Finally, we consider inference for linear regression:

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \mathbb{E}[\ell_\theta(X_i, Y_i)] = \arg \min_{\theta \in \mathbb{R}^d} \mathbb{E}[(Y_i - X_i^\top\theta)^2]. \quad (4.6)$$

While it is possible to obtain an algorithm for linear regression based on Theorem 15, one can derive a more powerful solution by using the fact that the natural estimator for problem (4.6) is linear in Y . We exploit these further properties in Algorithm 4 and Proposition 19, where we state a method for prediction-powered linear regression and establish its validity, respectively.

Proposition 19 (Linear regression). *Let θ^* be the linear regression solution (4.6) and fix $j^* \in [d]$. Then, the prediction-powered confidence interval in Algorithm 4 has valid coverage: $\liminf_{n,N \rightarrow \infty} P(\theta_{j^*}^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha$.*

Algorithm 1 Prediction-powered mean estimation

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , error level $\alpha \in (0, 1)$

- 1: $\hat{\theta}^{\text{PP}} \leftarrow \hat{\theta}^f - \hat{\Delta} := \frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i) - \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)$ ▷ prediction-powered estimator
- 2: $\hat{\sigma}_f^2 \leftarrow \frac{1}{N} \sum_{i=1}^N (f(\tilde{X}_i) - \hat{\theta}^f)^2$ ▷ empirical variance of imputed estimate
- 3: $\hat{\sigma}_{f-Y}^2 \leftarrow \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i - \hat{\Delta})^2$ ▷ empirical variance of empirical rectifier
- 4: $w_\alpha \leftarrow z_{1-\alpha/2} \sqrt{\frac{\hat{\sigma}_{f-Y}^2}{n} + \frac{\hat{\sigma}_f^2}{N}}$ ▷ normal approximation

Output: prediction-powered confidence set $\mathcal{C}_\alpha^{\text{PP}} = (\hat{\theta}^{\text{PP}} \pm w_\alpha)$

Algorithm 2 Prediction-powered quantile estimation

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , quantile $q \in (0, 1)$, error level $\alpha \in (0, 1)$

1: Construct fine grid Θ_{grid} between $\min_{i \in [N]} f(\tilde{X}_i)$ and $\max_{i \in [N]} f(\tilde{X}_i)$

2: **for** $\theta \in \Theta_{\text{grid}}$ **do**

3: $\hat{\Delta}_\theta \leftarrow \frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\})$ ▷ empirical rectifier

4: $\hat{F}(\theta) \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{f(\tilde{X}_i) \leq \theta\}$ ▷ imputed CDF

5: $\hat{\sigma}_{\hat{\Delta}}^2(\theta) \leftarrow \frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\} - \hat{\Delta}_\theta)^2$ ▷ empirical variance of empirical rectifier

6: $\hat{\sigma}_g^2(\theta) \leftarrow \frac{1}{N} \sum_{i=1}^N (\mathbb{1}\{f(\tilde{X}_i) \leq \theta\} - \hat{F}(\theta))^2$ ▷ empirical variance of imputed CDF

7: $w_\alpha(\theta) \leftarrow z_{1-\alpha/2} \sqrt{\frac{\hat{\sigma}_{\hat{\Delta}}^2(\theta)}{n} + \frac{\hat{\sigma}_g^2(\theta)}{N}}$ ▷ normal approximation

Output: prediction-powered confidence set $\mathcal{C}_\alpha^{\text{PP}} = \{\theta \in \Theta_{\text{grid}} : |\hat{F}(\theta) + \hat{\Delta}_\theta - q| \leq w_\alpha(\theta)\}$

Algorithm 3 Prediction-powered logistic regression

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , error level $\alpha \in (0, 1)$

1: Construct fine grid $\Theta_{\text{grid}} \subset \mathbb{R}^d$ of possible coefficients

2: $\hat{\Delta}_j \leftarrow \frac{1}{n} \sum_{i=1}^n X_{i,j} (f(X_i) - Y_i)$, $j \in [d]$ ▷ empirical rectifier

3: $\hat{\sigma}_{\hat{\Delta},j}^2 \leftarrow \frac{1}{n} \sum_{i=1}^n (X_{i,j} (f(X_i) - Y_i) - \hat{\Delta}_j)^2$, $j \in [d]$ ▷ empirical variance of empirical rectifier

4: **for** $\theta \in \Theta_{\text{grid}}$ **do**

5: $\hat{g}_{\theta,j}^f \leftarrow \frac{1}{N} \sum_{i=1}^N \tilde{X}_{i,j} (\mu_\theta(\tilde{X}_i) - f(\tilde{X}_i))$, $j \in [d]$, where $\mu_\theta(x) = \frac{1}{1 + \exp(-x^\top \theta)}$ ▷ imputed gradient

6: $\hat{\sigma}_{g,j}^2(\theta) \leftarrow \frac{1}{N} \sum_{i=1}^N (\tilde{X}_{i,j} (\mu_\theta(\tilde{X}_i) - f(\tilde{X}_i)) - \hat{g}_{\theta,j}^f)^2$, $j \in [d]$ ▷ empirical variance of imputed gradient

7: $w_{\alpha,j}(\theta) \leftarrow z_{1-\alpha/(2d)} \sqrt{\frac{\hat{\sigma}_{\hat{\Delta},j}^2}{n} + \frac{\hat{\sigma}_{g,j}^2(\theta)}{N}}$, $j \in [d]$ ▷ normal approximation

Output: prediction-powered confidence set $\mathcal{C}_\alpha^{\text{PP}} = \{\theta \in \Theta_{\text{grid}} : |\hat{g}_{\theta,j}^f + \hat{\Delta}_j| \leq w_{\alpha,j}(\theta), \forall j \in [d]\}$

Algorithm 4 Prediction-powered linear regression

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , coefficient $j^* \in [d]$, error level $\alpha \in (0, 1)$

- 1: $\hat{\theta}^{\text{PP}} \leftarrow \tilde{\theta}^f - \hat{\Delta} := \tilde{X}^\dagger f(\tilde{X}) - X^\dagger (f(X) - Y)$ \triangleright prediction-powered estimator
- 2: $\tilde{\Sigma} \leftarrow \frac{1}{N} \tilde{X}^\top \tilde{X}$, $\tilde{M} \leftarrow \frac{1}{N} \sum_{i=1}^N (f(\tilde{X}_i) - \tilde{X}_i^\top \tilde{\theta}^f)^2 \tilde{X}_i \tilde{X}_i^\top$
- 3: $\tilde{V} \leftarrow \tilde{\Sigma}^{-1} \tilde{M} \tilde{\Sigma}^{-1}$ \triangleright “sandwich” variance estimator for imputed estimate
- 4: $\Sigma \leftarrow \frac{1}{n} X^\top X$, $M \leftarrow \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i - X_i^\top \hat{\Delta})^2 X_i X_i^\top$
- 5: $V \leftarrow \Sigma^{-1} M \Sigma^{-1}$ \triangleright “sandwich” variance estimator for empirical rectifier
- 6: $w_\alpha \leftarrow z_{1-\alpha/2} \sqrt{\frac{V_{j^*j^*}}{n} + \frac{\tilde{V}_{j^*j^*}}{N}}$ \triangleright normal approximation

Output: prediction-powered confidence set $\mathcal{C}_\alpha^{\text{PP}} = (\hat{\theta}_{j^*}^{\text{PP}} \pm w_\alpha)$

Algorithm 5 Prediction-powered convex estimation

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , error level $\alpha \in (0, 1)$

- 1: Construct fine grid Θ_{grid}
- 2: **for** $\theta \in \Theta_{\text{grid}}$ **do**
- 3: $\hat{\Delta}_{\theta,j} \leftarrow \frac{1}{n} \sum_{i=1}^n (g_\theta(X_i, Y_i)_j - g_\theta(X_i, f(X_i))_j)$ \triangleright empirical rectifier
- 4: $\hat{g}_{\theta,j}^f \leftarrow \frac{1}{N} \sum_{i=1}^N g_\theta(\tilde{X}_i, f(\tilde{X}_i))_j$ \triangleright imputed gradient
- 5: $\hat{\sigma}_\Delta^2(\theta) \leftarrow \frac{1}{n} \sum_{i=1}^n (g_\theta(X_i, Y_i)_j - g_\theta(X_i, f(X_i))_j - \hat{\Delta}_\theta)^2$ \triangleright empirical variance of empirical rectifier
- 6: $\hat{\sigma}_g^2(\theta) \leftarrow \frac{1}{N} \sum_{i=1}^N (g_\theta(\tilde{X}_i, f(\tilde{X}_i))_j - \hat{g}_{\theta,j}^f)^2$ \triangleright empirical variance of imputed gradient
- 7: $w_\alpha(\theta) \leftarrow z_{1-\alpha/2} \sqrt{\frac{\hat{\sigma}_\Delta^2(\theta)}{n} + \frac{\hat{\sigma}_g^2(\theta)}{N}}$ \triangleright normal approximation

Output: prediction-powered confidence set $\mathcal{C}_\alpha^{\text{PP}} = \{\theta \in \Theta_{\text{grid}} : |\hat{g}_{\theta,j}^f + \hat{\Delta}_\theta| \leq w_\alpha(\theta)\}$

Estimand	Prediction-based gradient \hat{g}_θ^f	Rectifier $\hat{\Delta}_\theta$	Procedure
Mean	$\theta - \frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i)$	$\frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)$	Alg. 1
Median	$\frac{1}{2N} \sum_{i=1}^N \text{sign}(\theta - f(\tilde{X}_i))$	$\frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{f(X_i) \leq \theta\} - \mathbb{1}\{Y_i \leq \theta\})$	Alg. 2
q -quantile	$-q + \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{f(\tilde{X}_i) \leq \theta\}$	$\frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{f(X_i) \leq \theta\} - \mathbb{1}\{Y_i \leq \theta\})$	Alg. 2
Logistic regression	$\frac{1}{N} \sum_{i=1}^N \tilde{X}_i^T \left(\frac{1}{1+e^{-\theta^T \tilde{X}_i}} - f(\tilde{X}_i) \right)$	$\frac{1}{n} \sum_{i=1}^n X_i (f(X_i) - Y_i)$	Alg. 3
Linear regression	$\theta - \tilde{X}^T f(\tilde{X})$	$X^+ (f(X) - Y)$	Alg. 4
Convex minimizer	$\frac{1}{N} \sum_{i=1}^N \nabla \ell_\theta(\tilde{X}_i, f(\tilde{X}_i))$	$\frac{1}{n} \sum_{i=1}^n (\nabla \ell_\theta(X_i, f(X_i)) - \nabla \ell_\theta(X_i, Y_i))$	Alg. 5

Table 4.1: Prediction-powered inference for common statistical problems.

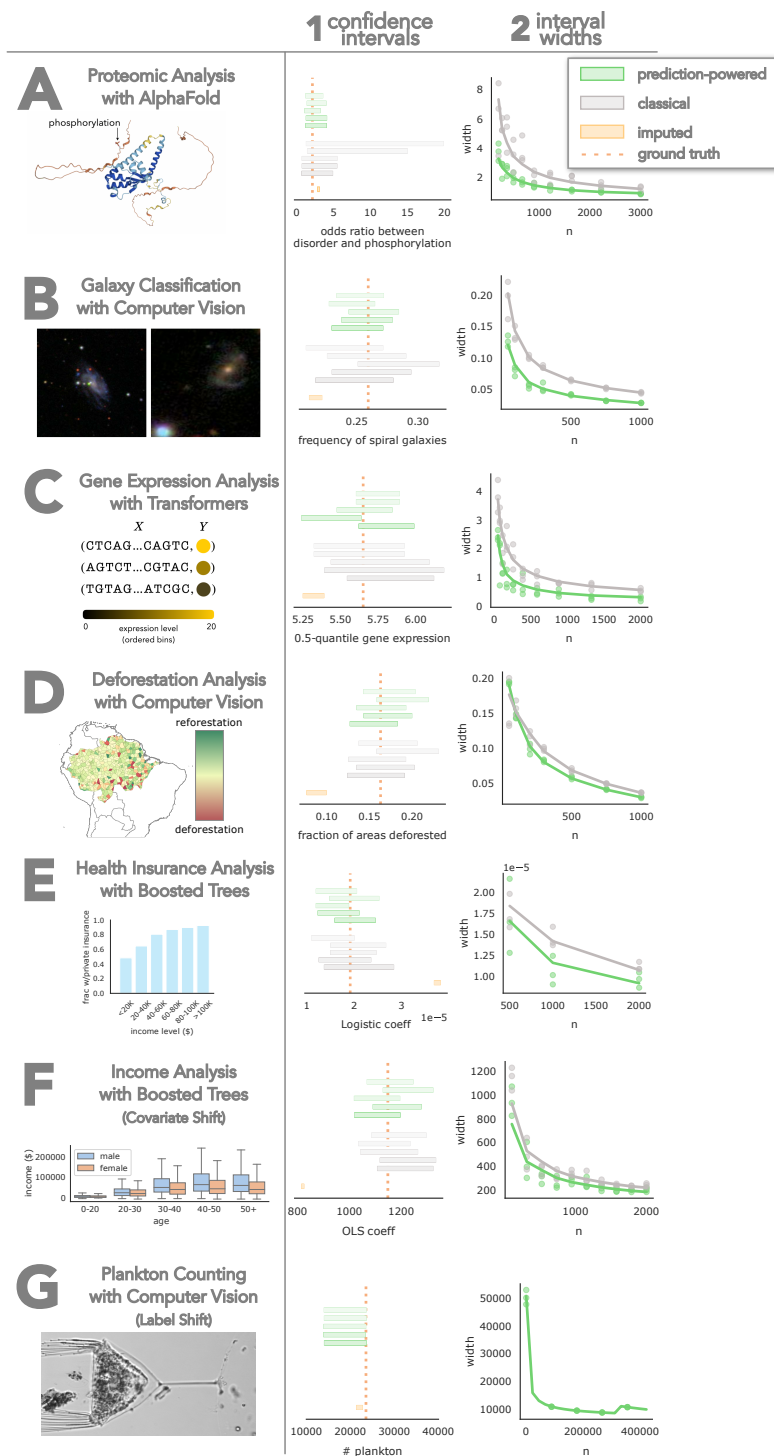



Figure 4.1: Comparison of prediction-powered, classical, and imputation approaches. Each row (A-G) is a different application. Panel (1) plots five randomly chosen intervals and (2) plots the width for varying n .

Problem	Prediction-powered	Classical
A Proteomic analysis with AlphaFold	$n = 316$	$n = 799$
B Galaxy classification with computer vision	$n = 189$	$n = 449$
C Gene expression analysis with transformers	$n = 764$	$n = 900$
D Deforestation analysis with computer vision	$n = 21$	$n = 35$
E Health insurance analysis with boosted trees	$n = 5569$	$n = 6653$
F Income analysis with boosted trees	$n = 177$	$n = 282$

Table 4.2: Number of labeled examples needed to make a discovery with prediction-powered inference and classical inference. The rows (A to F) correspond to the application domains from Figure 4.1. For each application, a null hypothesis about θ^* is tested at level 95%.

4.3 Applications

We demonstrate prediction-powered inference on real tasks. In each, we compute a prediction-powered confidence interval for an estimand and compare it to intervals obtained through the classical approach and the imputation approach. In all cases, we show that the imputation approach, which uses machine-learning predictions without accounting for prediction errors, does not contain the true value of the estimand. We compare the widths of the two valid approaches, prediction-powered and classical, as a function of the amount of labeled data used. In addition, we compare the number of labeled examples needed to reject a null hypothesis at level $1 - \alpha = 95\%$ with high probability. Each trial randomly splits the data into a labeled dataset and an unlabeled dataset. The results are given in Figure 4.1 and Table 4.2.

See [8] for a Python package implementing prediction-powered inference, which contains code for reproducing the experiments. Each application below comes with a corresponding Jupyter notebook that can be accessed by clicking these icons: . We packaged the data in such a way that the reader can run the notebooks on their local machine without downloading large datasets.

4.3.1 Relating protein structure and post-translational modifications

The goal is to characterize whether various types of post-translational modifications (PTMs) occur more frequently in intrinsically disordered regions (IDRs) of proteins [86]. Recently, Bludau et al. [25] studied this relationship on an unprecedented proteome-wide scale by using structures predicted by AlphaFold [89] to predict IDRs, in contrast to previous work which was limited to far fewer experimentally derived structures.

To quantify the association between PTMs and IDRs, the authors applied the impu-

tation approach: they computed the odds ratio between AlphaFold-based IDR predictions and PTMs on a dataset of hundreds of thousands of protein sequence residues [149]. Using prediction-powered inference, we can combine AlphaFold-based predictions together with gold-standard IDR labels to give a confidence interval for the true odds ratio that is statistically valid, in contrast with the interval constructed with the imputation approach, and smaller than the interval constructed using the classical approach.

We use the fact that the odds ratio, θ^* , between whether or not a protein residue is part of an IDR, $Y \in \{0, 1\}$, and whether or not it has a PTM, $Z \in \{0, 1\}$, can be written as a function of two means:

$$\theta^* = \frac{\mu_1/(1 - \mu_1)}{\mu_0/(1 - \mu_0)}, \quad (4.7)$$

where $\mu_1 = P(Y = 1 | Z = 1)$ and $\mu_0 = P(Y = 1 | Z = 0)$. We therefore proceed by constructing $1 - \alpha/2$ prediction-powered confidence intervals for μ_0 and μ_1 , denoted $\mathcal{C}_0^{\text{PP}} = [l_0, u_0]$ and $\mathcal{C}_1^{\text{PP}} = [l_1, u_1]$, respectively. We then propagate $\mathcal{C}_0^{\text{PP}}$ and $\mathcal{C}_1^{\text{PP}}$ through the odds-ratio formula (4.7) to get the following confidence interval:

$$\mathcal{C}^{\text{PP}} = \left\{ \frac{c_1}{1 - c_1} \cdot \frac{1 - c_0}{c_0} : c_0 \in \mathcal{C}_0^{\text{PP}}, c_1 \in \mathcal{C}_1^{\text{PP}} \right\} = \left(\frac{l_1}{1 - l_1} \cdot \frac{1 - u_0}{u_0}, \frac{u_1}{1 - u_1} \cdot \frac{1 - l_0}{l_0} \right).$$

By a union bound, \mathcal{C}^{PP} contains θ^* with probability at least $1 - \alpha$.

We have 10803 data points from Bludau et al. [25]. For each of 100 trials, we randomly sample n points to serve as the labeled dataset and treated the remaining $N = 10803 - n$ points as the unlabeled dataset for which we do not observe the IDR labels. For all values of n , the prediction-powered confidence intervals were smaller than classical intervals; see row A in Figure 4.1. Often, the classical intervals were large enough that they contained the odds ratio value of one, which means the direction of the association could not be determined from the confidence interval. On the other hand, the imputed confidence interval was far too small and significantly overestimated the true odds ratio. To reject the null hypothesis that the odds ratio is no greater than one, prediction-powered inference required $n = 316$ labeled observations, and the classical approach required $n = 799$ labeled observations; see row A in Table 4.2.

4.3.2 Galaxy classification

The goal is to determine the demographics of galaxies with spiral arms, which are correlated with star formation in the discs of low-redshift galaxies, and therefore, contribute to the understanding of star formation in the Local Universe. A large citizen science initiative called Galaxy Zoo 2 [165] has collected human annotations of roughly 300000 images of galaxies from the Sloan Digital Sky Survey [170] with the goal of measuring these demographics. We seek to explore the use of machine learning to improve the effective sample size and decrease the requisite number of human-annotated galaxies.

We focus on estimating the fraction of galaxies with spiral arms. We have 1364122 labeled galaxy images from Galaxy Zoo 2, from which we simulate labeled and unlabeled datasets as follows. For each of 100 trials, we randomly sample n points to serve as the labeled dataset and use the remaining $N = 1364122 - n$ points as the unlabeled dataset. We then use the algorithm for prediction-powered mean estimation to construct intervals. The prediction-powered confidence intervals for the mean are consistently much smaller than the classical intervals while retaining validity, and the imputation strategy fails to cover; see Figure 4.1, row B. To reject the null hypothesis that the fraction of galaxies with spiral arms is at most 0.2, prediction-powered inference requires $n = 189$ labeled examples, and classical inference requires $n = 449$ examples; see Table 4.2, row B.

4.3.3 Distribution of gene expression levels

Next, we construct prediction-powered confidence intervals on quantiles that characterize how a population of promoter sequences affects gene expression. Recently, Vaishnav et al. [150] trained a state-of-the-art transformer model to predict the expression level of a particular gene induced by a promoter sequence. They used the model’s predictions to study the effects of promoters—for example, by assessing how quantiles of predicted expression levels differ between different populations of promoters.

Here we focus on estimating different quantiles of gene expression levels induced by native yeast promoters. We have 61150 labeled native yeast promoter sequences from Vaishnav et al. [150], from which we simulate labeled and unlabeled datasets as follows. For each of 100 trials, we randomly sample n points to serve as the labeled dataset and use the remaining $N = 61150 - n$ points as the unlabeled dataset. We then use the second and third row of Table 1 to construct prediction-powered intervals for the median, as well as the 25%- and 75%-quantiles, of the expression levels. The prediction-powered confidence intervals for all three quantiles are much smaller than the classical intervals for all values of n . See row C in Figure 4.1 for the results for the median, and Figure 4.7 in Appendix 4.5.35 for the other two quantiles. We also evaluate the number of labeled examples required by prediction-powered inference and classical inference, respectively, to reject the null hypothesis that the median gene expression level is at most five. Prediction-powered inference requires $n = 764$ examples and classical inference requires $n = 900$ examples; see row C in Table 4.2.

4.3.4 Estimating deforestation in the Amazon

The goal is to estimate the fraction of the Amazon rainforest lost between 2000 and 2015. Gold-standard deforestation labels for parcels of land are scarce, having been collected largely through field visits, an expensive process impractical for large areas [32]. However, machine-learning predictions of forest cover based on satellite imagery are readily available for the entire Amazon [140].

We begin with 1596 gold-standard deforestation labels for parcels of land in the Amazon. For each of 100 trials, we randomly sample n data points to serve as the labeled dataset and use the remaining data points as the unlabeled dataset. We use the first row of Table 1 to construct the prediction-powered intervals. The imputation approach yields a small confidence interval that fails to cover the true deforestation fraction. The classical approach does cover the truth at the expense of a wider interval and, accordingly, diminished inferential power. The prediction-powered intervals are smaller than the classical intervals and retain validity; see row D in Figure 4.1. We also compare the number of gold-standard deforestation labels required by prediction-powered inference and the classical approach to reject the null hypothesis that there is no deforestation. We obtain $n = 21$ labels for prediction-powered inference and $n = 35$ labels for the classical approach; see row D in Table 4.2.

4.3.5 Relationship between income and private health insurance



The goal is to investigate the quantitative effect of income on the procurement of private health insurance using US census data. Concretely, we use the Folktables interface [57] to download census data from California in the year 2019 (378817 individuals).

As the labeled dataset with the health insurance indicator, we randomly sample n census entries. The remaining data is used as the unlabeled dataset. We use a gradient-boosted tree [46] trained on the previous year's data to predict the health insurance indicator in 2019. We construct a prediction-powered confidence interval on the logistic regression coefficient using the fifth row of Table 4.1. Results in row E in Figure 4.1 show that prediction-powered inference covers the ground truth, the classical interval is wider, and the imputation strategy fails to cover. We also compare the number of gold-standard labels required by prediction-powered inference and the classical approach to reject the null hypothesis that the logistic regression coefficient is no greater than $1.5 \cdot 10^{-5}$. We observe a significant sample size reduction with prediction-powered inference, which requires $n = 5569$ labels, whereas classical inference requires $n = 6653$ labels.

4.3.6 Relationship between age and income in a covariate-shifted population



The goal is to investigate the relationship between age and income using US census data. We use the same dataset as in the previous experiment, but the features are age and sex, and the target is yearly income in dollars. Furthermore, we introduce a shift in the distribution of the covariates between the gold-standard and unlabeled datasets by randomly sampling the unlabeled dataset with sampling weights 0.8 for females and 0.2 for males.

We used a gradient-boosted tree [46] trained on the previous year's raw data to predict the income in 2019. We construct a prediction-powered confidence interval on the ordinary least

squares regression coefficient using the covariate-shift-robust version of prediction-powered inference from Corollary 21. Results in row F in Figure 4.1 show that prediction-powered inference covers the ground truth, the classical interval is wider, and the imputation strategy fails to cover. We also compare the number of gold-standard labels required by prediction-powered inference and the classical approach to reject the null hypothesis that the OLS regression coefficient is no greater than 800. We observe a significant sample size reduction with prediction-powered inference, which requires $n = 177$ labels, whereas classical inference requires $n = 282$ labels.

4.3.7 Counting plankton

Assessment of the increases in phytoplankton growth during springtime warming is important for the study of global biogeochemical cycling in response to climate change. We counted the number of plankton observed by the Imaging FlowCytobot [118, 119], an automated, submersible flow cytometry system, at Woods Hole Oceanographic Institution in the year 2014. We have access to data from 2013, which are labeled, and we impute the 2014 data with machine-learning predictions from a state-of-the-art ResNet fine-tuned on all data up to and including 2012. The X_i are images of organic matter taken by the FlowCytobot and the Y_i are one of $\{\text{detritus}, \text{plankton}\}$, where **detritus** represents unspecified organic matter. The labeled dataset consist of 421238 image-label pairs from 2013 and we receive 329832 labeled images from 2014. We use the data from 2014 as our unlabeled data and confirm our results against those that were hand-labeled. The years 2013 and 2014 have a distribution shift, primarily caused by the change in the base frequency of plankton observations with respect to detritus. To apply prediction-powered inference to count the number of plankton recorded in 2014, we use the label-shift-robust technique described in Theorem 22. The results in row G in Figure 4.1 show that prediction-powered inference covers the ground truth and the imputation strategy fails to cover.

4.4 Extensions

We demonstrate that the framework of prediction-powered inference is applicable beyond inference under i.i.d. observations and convex losses studied in Section 4.2. First, we provide a strategy for prediction-powered inference when θ^* can be expressed as the optimum of any optimization problem, not necessarily a convex one. Then, we discuss prediction-powered inference under certain forms of distribution shift. We end with a brief discussion of a natural estimation strategy suggested by prediction-powered inference.

4.4.1 Beyond convex estimation

The tools developed in Section 4.2 were tailored to unconstrained convex optimization problems. In general, however, inferential targets can be defined in terms of nonconvex losses

or they may have (possibly even nonconvex) constraints. For such general optimization problems, we cannot expect the condition (4.1) to hold. In this section we generalize our approach to a broad class of risk minimizers:

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{E}[\ell_\theta(X_i, Y_i)], \quad (4.8)$$

where $\ell_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a possibly nonconvex loss function and Θ is an arbitrary set of admissible parameters. As before, if θ^* is not a unique minimizer, our method will return a set that contains all minimizers.

The problem (4.8) subsumes all previously studied settings. Indeed, when the loss ℓ_θ is convex and subdifferentiable and $\Theta = \mathbb{R}^p$ for some p —which is the case for all problems previously studied— θ^* can be equivalently characterized via the condition (4.1). In this section we provide a solution that can handle problems of the form (4.8) in full generality. We note, however, that the solution does not reduce to the one in Section 4.2 for convex estimation problems, and we expect the method from Section 4.2 to be more powerful for convex estimation problems with low-dimensional rectifiers.

To correct the imputation approach, we rely on the following rectifier:

$$\Delta_\theta = \mathbb{E}[\ell_\theta(X_i, Y_i) - \ell_\theta(X_i, f(X_i))]. \quad (4.9)$$

Notice that the rectifier (4.9) is always one-dimensional, while the rectifier (4.2) was p -dimensional.

One key difference relative to the approach of Section 4.2 is that we have an additional step of data splitting. We need the additional step because, unlike in convex estimation where we know $\mathbb{E}[g_{\theta^*}(X_i, Y_i)] = 0$, for general problems we do not know the value of $\mathbb{E}[\ell_{\theta^*}(X_i, Y_i)]$. To circumvent this issue, we estimate $\mathbb{E}[\ell_{\theta^*}(X_i, Y_i)]$ by approximating θ^* with an imputed estimate on the first $N/2$ unlabeled data points (for simplicity, take N to be even). To state the main result, we define

$$\tilde{\theta}^f = \arg \min_{\theta \in \Theta} \frac{2}{N} \sum_{i=1}^{N/2} \ell_\theta(\tilde{X}_i, f(\tilde{X}_i)), \quad \tilde{L}^f(\theta) := \frac{2}{N} \sum_{i=N/2+1}^N \ell_\theta(\tilde{X}_i, f(\tilde{X}_i)).$$

Theorem 20 (General risk minimization). *Fix $\alpha \in (0, 1)$ and $\delta \in (0, \alpha)$. Suppose that, for any $\theta \in \Theta$, we can construct $(\mathcal{R}_{\delta/2}^l(\theta), \mathcal{R}_{\delta/2}^u(\theta))$ and $(\mathcal{T}_{\frac{\alpha-\delta}{2}}^l(\theta), \mathcal{T}_{\frac{\alpha-\delta}{2}}^u(\theta))$ such that*

$$P(\Delta_\theta \leq \mathcal{R}_{\delta/2}^u(\theta)) \geq 1 - \delta/2; \quad P(\Delta_\theta \geq \mathcal{R}_{\delta/2}^l(\theta)) \geq 1 - \delta/2;$$

$$P(\tilde{L}^f(\theta) - \mathbb{E}[\ell_\theta(X_i, f(X_i))] \leq \mathcal{T}_{\frac{\alpha-\delta}{2}}^u(\theta)) \geq 1 - \frac{\alpha-\delta}{2}; \quad P(\tilde{L}^f(\theta) - \mathbb{E}[\ell_\theta(X_i, f(X_i))] \geq \mathcal{T}_{\frac{\alpha-\delta}{2}}^l(\theta)) \geq 1 - \frac{\alpha-\delta}{2}.$$

Let

$$\mathcal{C}_\alpha^{\text{PP}} = \left\{ \theta \in \Theta : \tilde{L}^f(\theta) \leq \tilde{L}^f(\tilde{\theta}^f) - \mathcal{R}_{\delta/2}^l(\theta) + \mathcal{R}_{\delta/2}^u(\tilde{\theta}^f) + \mathcal{T}_{\frac{\alpha-\delta}{2}}^u(\theta) - \mathcal{T}_{\frac{\alpha-\delta}{2}}^l(\tilde{\theta}^f) \right\}.$$

Then, we have

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

For example, if the loss $\ell_\theta(x, y)$ takes values in $[0, B]$ for all x, y , then we can set $\mathcal{T}_{\alpha-\delta}(\theta) = B\sqrt{\frac{\log(1/(\alpha-\delta))}{N}}$. The validity of this choice follows by Hoeffding's inequality.

Mode estimation. A commonplace inference task that does not fall under convex estimation is the problem of estimating the mode of the outcome distribution. When the outcome takes values in a discrete set Θ , this can be done by using the loss function $\ell_\theta(y) = \mathbb{1}\{y \neq \theta\}$, $\theta \in \Theta$. A generalization of this approach to continuous outcome distributions is obtained by defining the loss $\ell_\theta(y) = \mathbb{1}\{|y - \theta| > \eta\}$, for some width parameter $\eta > 0$. The target of inference is thus the point $\theta \in \mathbb{R}$ that has the most probability mass in its η -neighborhood, $\theta^* = \arg \min_{\theta \in \mathbb{R}} P(|Y_i - \theta| > \eta)$. Theorem 20 applies directly in both the discrete and continuous cases.

Tukey's biweight robust mean. The Tukey biweight loss function is a commonly used loss in robust statistics that results in an outlier-robust mean estimate. It behaves approximately like a quadratic near the origin and is constant far away from the origin. Formally, Tukey's biweight loss function is given by

$$\ell_\theta(y) = \begin{cases} \frac{c^2}{6} \left(1 - \left(1 - \frac{(y-\theta)^2}{c^2}\right)^3\right), & |y - \theta| \leq c, \\ \frac{c^2}{6}, & \text{otherwise,} \end{cases}$$

where c is a user-specified tuning parameter. It is not hard to see that the function $\ell_\theta(y)$ is nonconvex and hence not amenable to the analysis in Section 4.2; however, Theorem 20 applies.

Model selection. Nonconvex risk minimization problems are ubiquitous in model selection. For example, a common model selection strategy is best subset selection, which optimizes the squared loss, $\ell_\theta(x, y) = (y - x^\top \theta)^2$, subject to the constraint $\Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq k\}$. Here, Θ is the space of all k -sparse vectors for a user-chosen parameter k . Even though the loss function is convex, Θ is a nonconvex constraint set and hence we cannot rely on the condition (4.1) to find the minimizer. However, Theorem 20 still applies.

4.4.2 Inference under distribution shift

In Section 4.2 we focused on forming prediction-powered confidence intervals when the labeled and unlabeled data come from the same distribution. Herein, we extend our tools to the case where the labeled data (X, Y) comes from \mathbb{P} and the unlabeled data (\tilde{X}, \tilde{Y}) —which defines the target of inference θ^* —comes from \mathbb{Q} , and these are related by either a label

shift or a covariate shift. For covariate shift, we handle all estimation problems previously studied; for label shift, we handle certain types of linear problems.

We will write $\mathbb{E}_{\mathbb{Q}}, \mathbb{E}_{\mathbb{P}}$, etc to indicate which distribution the data inside the expectation is sampled from.

Covariate shift

First, we assume that \mathbb{Q} is a known *covariate shift* of \mathbb{P} . That is, if we denote by $\mathbb{Q} = \mathbb{Q}_X \cdot \mathbb{Q}_{Y|X}$ and $\mathbb{P} = \mathbb{P}_X \cdot \mathbb{P}_{Y|X}$ the relevant marginal and conditional distributions, we assume that $\mathbb{Q}_{Y|X} = \mathbb{P}_{Y|X}$. As in previous sections, we consider estimands of the form

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}}[\ell_{\theta}(X_i, Y_i)]. \quad (4.10)$$

Estimands of the form (4.10) can be related to risk minimizers on \mathbb{P} using the Radon-Nikodym derivative. In particular, suppose that \mathbb{Q}_X is dominated by \mathbb{P}_X and assume that the Radon-Nikodym derivative $w(x) = \frac{\mathbb{Q}_X}{\mathbb{P}_X}(x)$ is known. Then, we can rewrite (4.10) as

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{E}_{\mathbb{P}}[\ell_{\theta}^w(X_i, Y_i)],$$

where $\ell_{\theta}^w(x, y) = w(x)\ell_{\theta}(x, y)$. In words, risk minimizers on \mathbb{Q} can simply be written as risk minimizers on \mathbb{P} , but with a reweighted loss function. This permits inference on the rectifier to be based on data sampled from \mathbb{P} as before. For concreteness, we explain the approach in detail for convex risk minimizers. Let

$$\Delta_{\theta}^w = \mathbb{E}_{\mathbb{P}}[g_{\theta}^w(X_i, Y_i) - g_{\theta}^w(X_i, f(X_i))],$$

where $g_{\theta}^w(x, y) = g_{\theta}(x, y) \cdot w(x)$ and g_{θ} is a subgradient of ℓ_{θ} as before. A confidence set for the above rectifier suffices for prediction-powered inference on θ^* .

Corollary 21 (Covariate shift). *Suppose that the problem (4.10) is a nondegenerate convex estimation problem. Fix $\alpha \in (0, 1)$ and $\delta \in (0, \alpha)$. Suppose that, for any $\theta \in \mathbb{R}^p$, we can construct $\mathcal{R}_{\delta}(\theta)$ and $\mathcal{T}_{\alpha-\delta}(\theta)$ satisfying*

$$P(\Delta_{\theta}^w \in \mathcal{R}_{\delta}(\theta)) \geq 1 - \delta; \quad P(\mathbb{E}[g_{\theta}^w(X_i, f(X_i))] \in \mathcal{T}_{\alpha-\delta}(\theta)) \geq 1 - (\alpha - \delta).$$

Let $\mathcal{C}_{\alpha}^{\text{PP}} = \{\theta : 0 \in \mathcal{R}_{\delta}(\theta) + \mathcal{T}_{\alpha-\delta}(\theta)\}$, where $+$ denotes the Minkowski sum. Then,

$$P(\theta^* \in \mathcal{C}_{\alpha}^{\text{PP}}) \geq 1 - \alpha.$$

The same reweighting principle can be used to handle nonconvex risk minimizers as in Section 4.4.1.

Label shift

Next, we analyze classification problems where the proportions of the classes in the labeled data is different from those in the unlabeled data. This problem has been studied before in the literature on domain adaptation, e.g. by Lipton et al. [106], but our treatment focuses on the formation of confidence intervals. Formally, let $\mathcal{Y} = \{1, \dots, K\}$ be the label space and assume that $\mathbb{Q}_{X|Y} = \mathbb{P}_{X|Y}$. We consider estimands of the form

$$\theta^* = \mathbb{E}_{\mathbb{Q}_Y}[\nu(Y)],$$

where $\nu : \mathcal{Y} \rightarrow \mathbb{R}$ is a fixed function. For example, choosing $\nu(y) = \mathbb{1}\{y = k\}$ for some $k \in [K]$ asks for inference on the proportion of instances that belong to class k .

Using an analogous decomposition to the one for mean estimation, we can write

$$\theta^* = \mathbb{E}_{\mathbb{Q}_f}[\nu(f)] + (\mathbb{E}_{\mathbb{Q}_Y}[\nu(Y)] - \mathbb{E}_{\mathbb{Q}_f}[\nu(f)]) = \theta^f + \mathbf{\Delta},$$

where \mathbb{Q}_f denotes the distribution of $f(X)$, $X \sim \mathbb{Q}_X$. The quantity θ^f can be estimated using the unlabeled data from \mathbb{Q} and the model. Estimating the quantity $\mathbf{\Delta}$ using observations from \mathbb{P} will require leveraging the structure of the distribution shift. Central to our analysis will be the confusion matrix

$$\mathcal{K}_{j,l} = \mathbb{Q}(f(X) = j \mid Y = l), \quad j, l \in [K].$$

The label-shift assumption implies that $\mathcal{K}_{j,l} = \mathbb{P}(f(X) = j \mid Y = l)$, which can be estimated from labeled data sampled from \mathbb{P} . In particular, we estimate \mathcal{K} from the labeled data as

$$\widehat{\mathcal{K}}_{j,l} = \frac{1}{n(l)} \sum_{i=1}^n \mathbb{1}\{f(X_i) = j, Y_i = l\}, \quad \text{where } n(l) = \sum_{i=1}^n \mathbb{1}\{Y_i = l\}.$$

Similarly, we can estimate $\mathbb{Q}_f(k)$, $k \in [K]$ as

$$\widehat{\mathbb{Q}}_f(k) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{f(\tilde{X}_i) = k\}.$$

Treating \mathbb{Q}_f and \mathbb{Q}_Y as vectors, notice that we can write $\mathbb{Q}_f = \mathcal{K}\mathbb{Q}_Y$, and hence $\mathbb{Q}_Y = \mathcal{K}^{-1}\mathbb{Q}_f$. This leads to a natural estimate of \mathbb{Q}_Y , $\widehat{\mathbb{Q}}_Y = \widehat{\mathcal{K}}^{-1}\widehat{\mathbb{Q}}_f$. Below, we use these quantities to construct a prediction-powered confidence interval for $\theta^* = \mathbb{E}_{\mathbb{Q}_Y}[\nu(Y)]$.

Theorem 22 (Label shift). *Fix $\alpha \in (0, 1)$ and $\delta \in (0, \alpha)$. Let*

$$\mathcal{C}_\alpha^{\text{PP}} = \left(\mathbb{E}_{\widehat{\mathbb{Q}}_Y}[\nu(Y)] \pm \left(\max_{l,k \in [K]} \max_{p \in \mathcal{C}_{l,k}} |\widehat{\mathcal{K}}_{l,k} - p| + \sqrt{\frac{1}{2N} \log \frac{2}{\alpha - \delta}} \right) \right),$$

where

$$\mathcal{C}_{l,k} = \left\{ p : n(k)\widehat{\mathcal{K}}_{l,k} \in \left[F_{\text{Binom}(n(k),p)}^{-1} \left(\frac{\delta}{2K^2} \right), F_{\text{Binom}(n(k),p)}^{-1} \left(1 - \frac{\delta}{2K^2} \right) \right] \right\}$$

and $F_{\text{Binom}(n(k),p)}$ denotes the Binomial CDF. Then,

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

Naturally, the confidence interval becomes more conservative as the number of classes grows. Also, the power of the bound depends on the smallest number of instances observed for a particular class.

4.4.3 Prediction-powered point estimate

Prediction-powered inference suggests a natural approach to constructing point estimates as well. Define the *rectified* loss function as

$$L^{\text{PP}}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell_{\theta}(\tilde{X}_i, f(\tilde{X}_i)) + \frac{1}{n} \sum_{i=1}^n (\ell_{\theta}(X_i, Y_i) - \ell_{\theta}(X_i, f(X_i))).$$

The expected value of the rectified loss is equal to the true population loss that θ^* minimizes: $\mathbb{E}[L^{\text{PP}}(\theta)] = \mathbb{E}[\ell_{\theta}(X_i, Y_i)]$. We define the prediction-powered point estimate as the minimizer of the rectified loss:

$$\hat{\theta}^{\text{PP}} = \arg \min_{\theta} L^{\text{PP}}(\theta).$$

The confidence intervals formed in Algorithms 1- 5 were implicitly based on the gradient of the rectified loss, $\nabla L^{\text{PP}}(\theta) = \hat{g}_{\theta}^f + \hat{\Delta}_{\theta}$. More precisely, they all used $\nabla L^{\text{PP}}(\theta)$ as a statistic for testing whether $\mathbb{E}[\nabla \ell_{\theta}(X_i, Y_i)] = 0$. Notice that the prediction-powered point estimate is always contained in the constructed confidence intervals, since it satisfies $\hat{g}_{\theta}^f + \hat{\Delta}_{\theta} = 0$.

4.5 Appendix for Prediction-Powered Inference

4.5.1 Prediction-powered p-values

By relying on the standard duality between confidence intervals and p-values, we can immediately repurpose the presented theory to compute valid prediction-powered p-values.

To formalize this, suppose that we want to test the hull hypothesis $H_0 : \theta^* \in \Theta_0$, for some set $\Theta_0 \in \mathbb{R}^p$ (for example, a common choice when $p = 1$ is $\Theta_0 = \mathbb{R}_{\leq 0}$). Let \mathcal{C}_{α} be a valid confidence interval. Then, we can construct a valid p-value as

$$P = \inf \{ \alpha : \theta_0 \notin \mathcal{C}_{\alpha}, \forall \theta_0 \in \Theta_0 \}.$$

A p-value P is valid if it is super-uniform under the null, meaning $P(P \leq u) \leq u$ for all $u \in [0, 1]$. This is indeed the case for the p-value defined above, because when $\theta^* \in \Theta_0$, we have

$$P(P \leq u) \leq P(\theta^* \notin \mathcal{C}_u) \leq u.$$

The first inequality follows by the definition of P and the fact that $\theta^* \in \Theta_0$, and the second inequality follows by the validity of \mathcal{C}_u at level $1 - u$. We are implicitly using the fact that $\mathcal{C}_u \subseteq \mathcal{C}_{u'}$ when $u \geq u'$.

The above derivation is a general recipe for deriving p-values from confidence intervals. For the prediction-powered confidence intervals stated in Algorithms 1-5 (derived from Theorem 33), the corresponding prediction-powered p-value is given by:

$$P^{\text{PP}} = \inf \left\{ \alpha : |\hat{g}_{\theta_0}^f + \hat{\Delta}_{\theta_0}| > w_\alpha(\theta_0), \forall \theta_0 \in \Theta_0 \right\}.$$

Below we state analogues of Algorithms 1-4 when the goal is to compute a prediction-powered p-value.

Algorithm 6 Prediction-powered p-value for the mean

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , null set Θ_0

- 1: $\hat{\theta}^{\text{PP}} \leftarrow \tilde{\theta}^f - \hat{\Delta} := \frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i) - \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i)$
- 2: $\hat{\sigma}_f^2 \leftarrow \frac{1}{N} \sum_{i=1}^N (f(\tilde{X}_i) - \tilde{\theta}^f)^2$
- 3: $\hat{\sigma}_{f-Y}^2 \leftarrow \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i - \hat{\Delta})^2$
- 4: Define $w_\alpha := z_{1-\alpha/2} \sqrt{\frac{\hat{\sigma}_{f-Y}^2}{n} + \frac{\hat{\sigma}_f^2}{N}}$

Output: prediction-powered p-value $P^{\text{PP}} = \inf \{ \alpha : \theta_0 \notin (\hat{\theta}^{\text{PP}} \pm w_\alpha), \forall \theta_0 \in \Theta_0 \}$

Algorithm 7 Prediction-powered p-value for the quantile

Input: labeled data (X, Y) , unlabeled features X' , predictor f , quantile q , null set Θ_0

- 1: **for** $\theta \in \Theta_0$ **do**
- 2: $\hat{\Delta}_\theta \leftarrow \frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\})$
- 3: $\hat{F}_\theta \leftarrow \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{f(\tilde{X}_i) \leq \theta\}$
- 4: $\hat{\sigma}_\Delta^2(\theta) \leftarrow \frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\} - \hat{\Delta}_\theta)^2$
- 5: $\hat{\sigma}_g^2(\theta) \leftarrow \frac{1}{N} \sum_{i=1}^N (\mathbb{1}\{f(\tilde{X}_i) \leq \theta\} - \hat{F}_\theta)^2$
- 6: Define $w_\alpha(\theta) := z_{1-\alpha/2} \sqrt{\frac{\hat{\sigma}_\Delta^2(\theta)}{n} + \frac{\hat{\sigma}_g^2(\theta)}{N}}$

Output: prediction-powered p-value $P^{\text{PP}} = \inf \{ \alpha : |\hat{F}_{\theta_0} + \hat{\Delta}_{\theta_0} - q| > w_\alpha(\theta_0), \forall \theta_0 \in \Theta_0 \}$

Algorithm 8 Prediction-powered p-value for logistic regression coefficients

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , null set Θ_0

- 1: $\hat{\Delta}_j \leftarrow \frac{1}{n} \sum_{i=1}^n X_{i,j} (f(X_i) - Y_i), \quad j \in [d]$
- 2: $\hat{\sigma}_{\Delta,j}^2 \leftarrow \frac{1}{n} \sum_{i=1}^n (X_{i,j} (f(X_i) - Y_i) - \hat{\Delta}_j)^2, \quad j \in [d]$
- 3: **for** $\theta \in \Theta_0$ **do**
- 4: $\hat{g}_{\theta,j}^f \leftarrow \frac{1}{N} \sum_{i=1}^N \tilde{X}_{i,j} (\mu_\theta(\tilde{X}_i) - f(\tilde{X}_i)), \quad j \in [d], \quad \text{where } \mu_\theta(x) = \frac{1}{1 + \exp(-x^\top \theta)}$
- 5: $\hat{\sigma}_{g,j}^2(\theta) \leftarrow \frac{1}{N} \sum_{i=1}^N (\tilde{X}_{i,j} (\mu_\theta(\tilde{X}_i) - f(\tilde{X}_i)) - \hat{g}_{\theta,j}^f)^2, \quad j \in [d]$
- 6: Define $w_{\alpha,j}(\theta) := z_{1-\alpha/(2d)} \sqrt{\frac{\hat{\sigma}_{\Delta,j}^2}{n} + \frac{\hat{\sigma}_{g,j}^2(\theta)}{N}}, \quad j \in [d]$

Output: prediction-powered p-value $P^{\text{PP}} = \inf \{ \alpha : |g_{\theta_0,j}^f + \hat{\Delta}_j| > w_{\alpha,j}(\theta_0), \forall j \in [d], \theta_0 \in \Theta_0 \}$

Algorithm 9 Prediction-powered p-value for linear regression coefficients**Input:** labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , coefficient j^* , null set Θ_0

- 1: $\hat{\theta}^{\text{PP}} \leftarrow \tilde{\theta}^f - \hat{\Delta} := \tilde{X}^\top f(\tilde{X}) - X^\top (f(X) - Y)$
- 2: $\tilde{\Sigma} \leftarrow \frac{1}{N} \tilde{X}^\top \tilde{X}$, $\tilde{M} \leftarrow \frac{1}{N} \sum_{i=1}^N (f(\tilde{X}_i) - \tilde{X}_i^\top \tilde{\theta}^f)^2 \tilde{X}_i \tilde{X}_i^\top$
- 3: $\tilde{V} \leftarrow (\tilde{\Sigma})^{-1} \tilde{M} (\tilde{\Sigma})^{-1}$
- 4: $\Sigma \leftarrow \frac{1}{n} X^\top X$, $M \leftarrow \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i - X_i^\top \hat{\Delta})^2 X_i X_i^\top$
- 5: $V \leftarrow \Sigma^{-1} M \Sigma^{-1}$

- 6: Define $w_\alpha := z_{1-\alpha/2} \sqrt{\frac{V_{j^*j^*}}{n} + \frac{\tilde{V}_{j^*j^*}}{N}}$

Output: prediction-powered confidence set $\mathcal{C}_\alpha^{\text{PP}} = \inf\{\alpha : \theta_0 \notin (\hat{\theta}_{j^*}^{\text{PP}} \pm w_\alpha), \forall \theta_0 \in \Theta_0\}$ **Corollary 23** (Mean p-value). *Let θ^* be the mean outcome:*

$$\theta^* = \mathbb{E}[Y_i].$$

Then, the prediction-powered p-value in Algorithm 6 is valid: under the null,

$$\liminf_{n, N \rightarrow \infty} P(P^{\text{PP}} \leq u) \leq u, \forall u \in [0, 1]$$

Corollary 24 (Quantile p-value). *Let θ^* be the q -quantile:*

$$\theta^* = \min\{\theta : P(Y_i \leq \theta) \geq q\}.$$

Then, the prediction-powered p-value in Algorithm 7 is valid: under the null,

$$\liminf_{n, N \rightarrow \infty} P(P^{\text{PP}} \leq u) \leq u, \forall u \in [0, 1]$$

Corollary 25 (Logistic regression p-value). *Let θ^* be the logistic regression solution:*

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \mathbb{E}[-Y_i \theta^\top X_i + \log(1 + \exp(\theta^\top X_i))].$$

Then, the prediction-powered p-value in Algorithm 8 is valid: under the null,

$$\liminf_{n, N \rightarrow \infty} P(P^{\text{PP}} \leq u) \leq u, \forall u \in [0, 1]$$

Corollary 26 (Linear regression p-value). *Fix $j^* \in [d]$. Let θ^* be the linear regression solution:*

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^d} \mathbb{E}[(Y_i - X_i^\top \theta)^2].$$

Then, the prediction-powered p-value in Algorithm 9 is valid: under the null,

$$\liminf_{n, N \rightarrow \infty} P(P^{\text{PP}} \leq u) \leq u, \forall u \in [0, 1]$$

4.5.2 Inference on a finite population

The techniques developed in this chapter directly translate to the *finite-population* setting. Here, we treat (\tilde{X}, \tilde{Y}) as a fixed finite population consisting of N feature-outcome pairs, without imposing any distributional assumptions on the data points. Analogously to the i.i.d. setting, we observe all features \tilde{X} and a small set of outcomes. Specifically, we assume that we observe $(\tilde{Y}_i)_{i \in \mathcal{I}}$, where $\mathcal{I} = \{i_1, \dots, i_n\}$ is a uniformly sampled subset of $[N]$ of size $n \ll N$. In this section we adapt all our main results to the finite-population context.

Given a loss function ℓ_θ and parameter space Θ , the target estimand is the risk minimizer we would compute if we could observe the whole population:

$$\theta^* = \arg \min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^N \ell_\theta(\tilde{X}_i, \tilde{Y}_i). \quad (4.11)$$

The following two subsections mirror the results for convex and nonconvex estimation from the main body of the chapter. All results in this section are proved essentially identically as their i.i.d. counterparts.

In what follows, we construct prediction-powered confidence sets $\mathcal{C}_\alpha^{\text{PP}}$ assuming a valid confidence set around the rectifier (defined below for the finite-population context). The confidence set for the rectifier can be constructed from $(\tilde{X}_i, \tilde{Y}_i)_{i \in \mathcal{I}}$ via a direct application of off-the-shelf results outlined in Appendix 4.5.21. In particular, in Proposition 40 we state an asymptotically valid interval for the mean based on a finite-population version of the central limit theorem, and in Proposition 39 we state a nonasymptotically valid interval for the mean for finite populations due to Waudby-Smith and Ramdas [161]. The only assumption required to apply the latter is that $g_\theta(\tilde{X}_i, \tilde{Y}_i) - g_\theta(\tilde{X}_i, f(\tilde{X}_i))$ has a known bound valid for all $i \in [N]$.

4.5.3 Convex estimation

In the finite-population setting, the mild nondegeneracy condition ensured by convexity takes the form

$$\frac{1}{N} \sum_{i=1}^N g_{\theta^*}(\tilde{X}_i, \tilde{Y}_i) = 0, \quad (4.12)$$

where g_θ is a subgradient of ℓ_θ . The rectifier is thus:

$$\Delta_\theta = \frac{1}{N} \sum_{i=1}^N (g_\theta(\tilde{X}_i, \tilde{Y}_i) - g_\theta(\tilde{X}_i, f(\tilde{X}_i))).$$

Theorem 27 (Convex estimation, finite population). *Suppose that the convex estimation problem is nondegenerate (4.12). Fix $\alpha \in (0, 1)$. Suppose that, for any $\theta \in \mathbb{R}^p$, we can construct $\mathcal{R}_\alpha(\theta)$ satisfying*

$$P(\Delta_\theta \in \mathcal{R}_\alpha(\theta)) \geq 1 - \alpha.$$

Let $\mathcal{C}_\alpha^{\text{PP}} = \{\theta : -\frac{1}{N} \sum_{i=1}^N g_\theta(\tilde{X}_i, f(\tilde{X}_i)) \in \mathcal{R}_\alpha(\theta)\}$. Then,

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

We apply Theorem 27 in the context of mean estimation, quantile estimation, logistic regression, and linear regression. The target estimand θ^* is defined as in (4.11) with the loss function chosen appropriately, as discussed in Section 4.2.1. We remark that, just like in the i.i.d. case, the analysis for linear regression follows a more refined approach, as in the proof of Proposition 19.

Corollary 28 (Mean estimation, finite population). *Let θ^* be the mean outcome. Fix $\alpha \in (0, 1)$. Suppose that, for any $\theta \in \mathbb{R}$, we can construct an interval $(\mathcal{R}_\alpha^l, \mathcal{R}_\alpha^u)$ such that $P(\Delta \in (\mathcal{R}_\alpha^l, \mathcal{R}_\alpha^u)) \geq 1 - \alpha$, where*

$$\Delta = \frac{1}{N} \sum_{i=1}^N (f(\tilde{X}_i) - \tilde{Y}_i).$$

Let

$$\mathcal{C}_\alpha^{\text{PP}} = \left(\frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i) - \mathcal{R}_\alpha^u, \frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i) - \mathcal{R}_\alpha^l \right).$$

Then,

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

Corollary 29 (Quantile estimation, finite population). *Let θ^* be the q -quantile. Fix $\alpha \in (0, 1)$. Suppose that, for any $\theta \in \mathbb{R}$, we can construct an interval $(\mathcal{R}_\alpha^l(\theta), \mathcal{R}_\alpha^u(\theta))$ such that $P(\Delta_\theta \in (\mathcal{R}_\alpha^l(\theta), \mathcal{R}_\alpha^u(\theta))) \geq 1 - \alpha$, where*

$$\Delta_\theta = \frac{1}{N} \sum_{i=1}^N (\mathbb{1}\{\tilde{Y}_i \leq \theta\} - \mathbb{1}\{f(\tilde{X}_i) \leq \theta\}).$$

Let

$$\mathcal{C}_\alpha^{\text{PP}} = \left\{ \theta \in \mathbb{R} : \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{f(\tilde{X}_i) \leq \theta\} \in (q - \mathcal{R}_\alpha^u(\theta), q - \mathcal{R}_\alpha^l(\theta)) \right\}.$$

Then,

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

Corollary 30 (Logistic regression, finite population). *Let θ^* be the logistic regression solution. Fix $\alpha \in (0, 1)$. Suppose that we can construct $\mathcal{R}_\alpha^l, \mathcal{R}_\alpha^u \in \mathbb{R}^d$ such that $P(\Delta_j \in (\mathcal{R}_{\alpha,j}^l, \mathcal{R}_{\alpha,j}^u), \forall j \in [d]) \geq 1 - \alpha$, where*

$$\Delta = \frac{1}{N} \sum_{i=1}^N \tilde{X}_i (f(\tilde{X}_i) - \tilde{Y}_i).$$

Let

$$\mathcal{C}_\alpha^{\text{PP}} = \left\{ \theta \in \mathbb{R}^d : \frac{1}{N} \sum_{i=1}^N \tilde{X}_{i,j} \left(f(\tilde{X}_i) - \frac{1}{1 + \exp(-\tilde{X}_i^\top \theta)} \right) \in (\mathcal{R}_{\alpha,j}^l, \mathcal{R}_{\alpha,j}^u), \forall j \in [d] \right\}.$$

Then,

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

Corollary 31 (Linear regression, finite population). *Let θ^* be the linear regression solution. Fix $\alpha \in (0, 1)$. Suppose that we can construct $\mathcal{R}_\alpha^l, \mathcal{R}_\alpha^u \in \mathbb{R}^d$ such that $P(\Delta_j \in (\mathcal{R}_{\alpha,j}^l, \mathcal{R}_{\alpha,j}^u), \forall j \in [d]) \geq 1 - \alpha$, where*

$$\Delta = \tilde{X}^\dagger (f(\tilde{X}) - \tilde{Y}).$$

Let

$$\mathcal{C}_\alpha^{\text{PP}} = (\tilde{X}^\dagger f(\tilde{X}) - \mathcal{R}_\alpha^u, \tilde{X}^\dagger f(\tilde{X}) - \mathcal{R}_\alpha^l).$$

Then,

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

4.5.4 Beyond convex estimation

We now consider general risk minimizers in the finite-population context. The rectifier is equal to:

$$\Delta_\theta = \frac{1}{N} \sum_{i=1}^N (\ell_\theta(\tilde{X}_i, \tilde{Y}_i) - \ell_\theta(\tilde{X}_i, f(\tilde{X}_i))).$$

Unlike in the i.i.d. setting, there is no need for data splitting because the imputed estimate is deterministic. We let:

$$\tilde{L}^f(\theta) := \frac{1}{N} \sum_{i=1}^N \ell_\theta(\tilde{X}_i, f(\tilde{X}_i)); \quad \tilde{\theta}^f = \arg \min_{\theta \in \Theta} \tilde{L}^f(\theta).$$

Theorem 32 (General risk minimization, finite population). *Fix $\alpha \in (0, 1)$. Suppose that, for any $\theta \in \Theta$, we can construct $(\mathcal{R}_{\alpha/2}^l(\theta), \mathcal{R}_{\alpha/2}^u(\theta))$ such that*

$$P(\Delta_\theta \leq \mathcal{R}_{\alpha/2}^u(\theta)) \geq 1 - \alpha/2; \quad P(\Delta_\theta \geq \mathcal{R}_{\alpha/2}^l(\theta)) \geq 1 - \alpha/2.$$

Let

$$\mathcal{C}_\alpha^{\text{PP}} = \left\{ \theta \in \Theta : \tilde{L}^f(\theta) \leq \tilde{L}^f(\tilde{\theta}^f) - \mathcal{R}_{\alpha/2}^l(\theta) + \mathcal{R}_{\alpha/2}^u(\tilde{\theta}^f) \right\}.$$

Then, we have

$$P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

4.5.5 Deferred theoretical details

We state an asymptotic counterpart of Theorem 15 that is used to prove the propositions in Section 4.2.1. Then, we provide nonasymptotically-valid counterparts of the algorithms in Section 4.2.1. Finally, we state the regularity conditions necessary for the guarantees presented in Section 4.2.1.

4.5.6 Asymptotic counterpart of Theorem 15

The following is an asymptotic counterpart of Theorem 15 that uses the central limit theorem in the confidence set construction. We note the error budget splitting used in Theorem 15 is in fact not necessary, but we believe that it facilitates exposition when presenting nonasymptotic guarantees. The asymptotic result below is stated without the splitting of the error budget. The proof is stated in Appendix 4.5.9.

Theorem 33 (Convex estimation: asymptotic version). *Suppose that the convex estimation problem is nondegenerate as in (4.1) and that $\frac{n}{N} \rightarrow p$, for some $p \in (0, 1)$. Fix $\alpha \in (0, 1)$. For all $\theta \in \mathbb{R}^p$, define*

$$\hat{\Delta}_\theta = \frac{1}{n} \sum_{i=1}^n (g_\theta(X_i, Y_i) - g_\theta(X_i, f(X_i))); \quad \hat{g}_\theta^f = \frac{1}{N} \sum_{i=1}^N g_\theta(\tilde{X}_i, f(\tilde{X}_i)).$$

Further, denoting by $g_{\theta,j}(x, y)$ the j -th coordinate of $g_\theta(x, y)$, let

$$\hat{\sigma}_{\Delta,j}^2(\theta) = \frac{1}{n} \sum_{i=1}^n (g_{\theta,j}(X_i, Y_i) - g_{\theta,j}(X_i, f(X_i)) - \hat{\Delta}_{\theta,j})^2; \quad \hat{\sigma}_{g,j}^2(\theta) = \frac{1}{N} \sum_{i=1}^N (g_{\theta,j}(\tilde{X}_i, f(\tilde{X}_i)) - \hat{g}_{\theta,j}^f)^2,$$

for all $j \in [p]$. Let $w_{\alpha,j}(\theta) = z_{1-\alpha/(2p)} \sqrt{\frac{\hat{\sigma}_{\Delta,j}^2(\theta)}{n} + \frac{\hat{\sigma}_{g,j}^2(\theta)}{N}}$ and

$$\mathcal{C}_\alpha^{\text{PP}} = \{\theta : |\hat{\Delta}_{\theta,j} + \hat{g}_{\theta,j}^f| \leq w_{\alpha,j}(\theta), \forall j \in [p]\}.$$

Then,

$$\liminf_{n, N \rightarrow \infty} P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha.$$

4.5.7 Algorithms with nonasymptotic validity

We state nonasymptotically-valid algorithms for prediction-powered mean estimation, quantile estimation, and logistic regression. Like the methods in Section 4.2.1, the algorithms rely on the abstract recipe from Theorem 15. The proofs of validity are included in Appendix 4.5.9.

The following algorithms rely on any off-the-shelf method for computing confidence intervals for the mean. We choose a variance-adaptive confidence interval for the mean due to Waudby-Smith and Ramdas [161], which we state in Algorithm 13. We opt to present this construction as the default nonasymptotic confidence interval for the mean because of its strong practical performance. The only assumption required to apply Algorithm 13 is that the observations are almost surely bounded within a known interval.

Algorithm 10 Prediction-powered mean estimation (nonasymptotic)

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , error levels $\alpha, \delta \in (0, 1)$, bound B

- 1: $(f_{\alpha-\delta}^l, f_{\alpha-\delta}^u) \leftarrow \text{MeanCI}\left(\{f(\tilde{X}_i)\}_{i=1}^N, \text{err} = \alpha - \delta, \text{range} = [0, B]\right)$
- 2: $(\mathcal{R}_\delta^l, \mathcal{R}_\delta^u) \leftarrow \text{MeanCI}\left(\{f(X_i) - Y_i\}_{i=1}^n, \text{err} = \delta, \text{range} = [-B, B]\right)$

Output: prediction-powered confidence set $\mathcal{C}_\alpha^{\text{PP}} = (f_{\alpha-\delta}^l - \mathcal{R}_\delta^u, f_{\alpha-\delta}^u - \mathcal{R}_\delta^l)$

Corollary 34 (Mean estimation). *Let θ^* be the mean outcome (4.3). Suppose that $Y_i, f(X_i) \in [0, B]$ almost surely. Then, the prediction-powered confidence set in Algorithm 10 has valid coverage: $P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha$.*

Algorithm 11 Prediction-powered quantile estimation (nonasymptotic)

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , quantile $q \in (0, 1)$, error levels $\alpha, \delta \in (0, 1)$

- 1: Construct fine grid Θ_{grid} between $\min_{i \in [N]} f(\tilde{X}_i)$ and $\max_{i \in [N]} f(\tilde{X}_i)$
- 2: **for** $\theta \in \Theta_{\text{grid}}$ **do**
- 3: $(\mathcal{R}_\delta^l(\theta), \mathcal{R}_\delta^u(\theta)) \leftarrow \text{MeanCI}\left(\{\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\}\}_{i=1}^n, \text{err} = \delta, \text{range} = [-1, 1]\right)$
- 4: $(\hat{F}_{\alpha-\delta}^l(\theta), \hat{F}_{\alpha-\delta}^u(\theta)) \leftarrow \text{MeanCI}\left(\{\mathbb{1}\{f(\tilde{X}_i) \leq \theta\}\}_{i=1}^N, \text{err} = \alpha - \delta, \text{range} = [0, 1]\right)$

Output: prediction-powered confidence set

$$\mathcal{C}_\alpha^{\text{PP}} = \{\theta \in \Theta_{\text{grid}} : q \in (\hat{F}_{\alpha-\delta}^l(\theta) + \mathcal{R}_\delta^l(\theta), \hat{F}_{\alpha-\delta}^u(\theta) + \mathcal{R}_\delta^u(\theta))\}$$

Corollary 35 (Quantile estimation). *Let θ^* be the q -quantile (4.4). Then, the prediction-powered confidence set in Algorithm 11 has valid coverage: $P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha$.*

Algorithm 12 Prediction-powered logistic regression (nonasymptotic)

Input: labeled data (X, Y) , unlabeled features \tilde{X} , predictor f , error levels $\alpha, \delta \in (0, 1)$, bound $\mathbf{B} = (B_j)_{j=1}^d$

- 1: Construct fine grid $\Theta_{\text{grid}} \subset \mathbb{R}^d$ of possible coefficients
- 2: $(\mathcal{R}_{\delta,j}^l, \mathcal{R}_{\delta,j}^u) \leftarrow \text{MeanCI}(\{X_{i,j}(f(X_i) - Y_i)\}_{i=1}^n, \text{err} = \delta, \text{range} = [-B_j, B_j]), j \in [d]$
- 3: **for** $\theta \in \Theta_{\text{grid}}$ **do**
- 4: $(g_{\alpha-\delta,j}^l(\theta), g_{\alpha-\delta,j}^u(\theta)) \leftarrow \text{MeanCI}(\{\tilde{X}_{i,j}(\mu_\theta(\tilde{X}_i) - f(\tilde{X}_i))\}_{i=1}^N, \text{err} = \frac{\alpha-\delta}{d}, \text{range} = [-B_j, B_j]), j \in [d]$,
- 5: where $\mu_\theta(x) = \frac{1}{1+\exp(-x^\top \theta)}$

Output: prediction-powered confidence set

$$\mathcal{C}_\alpha^{\text{PP}} = \{\theta \in \Theta_{\text{grid}} : 0 \in [g_{\alpha-\delta,j}^l(\theta) + \mathcal{R}_{\delta,j}^l, g_{\alpha-\delta,j}^u(\theta) + \mathcal{R}_{\delta,j}^u], \forall j \in [d]\}$$

Corollary 36 (Logistic regression). *Let θ^* be the logistic regression solution (4.5). Suppose that $|X_{1,j}| \leq B_j$ and $Y_i, f(X_i) \in [0, 1]$ almost surely. Then, the prediction-powered confidence set in Algorithm 12 has valid coverage: $P(\theta^* \in \mathcal{C}_\alpha^{\text{PP}}) \geq 1 - \alpha$.*

We note that there exists an analogous nonasymptotic algorithm for linear regression, however we do not recommend it in practice. The reason is that the refined (but asymptotic) analysis used to prove Proposition 19 shows that it is sufficient to analyze a one-dimensional rectifier, while directly invoking Theorem 15 would require analyzing a d -dimensional rectifier and thus yields more conservative intervals.

Algorithm 13 MeanCI (see Proposition 37)

Input: data points $\{Z_1, \dots, Z_n\}$, error level $\alpha \in (0, 1)$, range $[L, U]$ s.t. $Z_i \in [L, U]$

- 1: For all $i \in [n]$, let $Z_i \leftarrow (Z_i - L)/(U - L)$ ▷ normalize data to interval $[0, 1]$
- 2: Construct fine grid M_{grid} of interval $[0, 1]$
- 3: Initialize active set $\mathcal{A} = M_{\text{grid}}$
- 4: **for** $t \in 1, \dots, n$ **do**
- 5: Set $\hat{\mu}_t \leftarrow \frac{0.5 + \sum_{j=1}^t Z_j}{t+1}$, $\hat{\sigma}_t^2 \leftarrow \frac{0.25 + \sum_{j=1}^t (Z_j - \hat{\mu}_t)^2}{t+1}$, $\lambda_t \leftarrow \sqrt{\frac{2 \log(2/\alpha)}{n \hat{\sigma}_{t-1}^2}}$
- 6: **for** $m \in \mathcal{A}$ **do**
- 7: $M_t^+(m) \leftarrow (1 + \min(\lambda_t, \frac{0.5}{m})(Z_t - m)) M_{t-1}^+(m)$
- 8: $M_t^-(m) \leftarrow (1 - \min(\lambda_t, \frac{0.5}{1-m})(Z_t - m)) M_{t-1}^-(m)$
- 9: $M_t(m) \leftarrow \frac{1}{2} \max\{M_t^+(m), M_t^-(m)\}$ ▷ construct test martingale for $m \in [0, 1]$
- 10: **if** $M_t(m) \geq 1/\alpha$ **then**
- 11: $\mathcal{A} \leftarrow \mathcal{A} \setminus \{m\}$ ▷ Remove m from active set

Output: Confidence set for the mean $\mathcal{C}_\alpha = \{m(U - L) + L : m \in \mathcal{A}\}$

4.5.8 Regularity conditions

All algorithms stated in Section 4.2 rely on confidence intervals derived from the central limit theorem. For such intervals to be asymptotically valid, we require that the two quantities whose mean is being estimated, namely $g_\theta(X_i, Y_i) - g_\theta(X_i, f(X_i))$ and $g_\theta(X_i, f(X_i))$, have at least the first two moments (see Proposition 38).

For Proposition 19 to hold, we need the same conditions as those required for classical linear regression intervals to cover the target. We note that these conditions are very weak; in particular, it is *not* required that the true data-generating process be linear or the errors be homoskedastic. See Buja et al. [30] for a detailed discussion. The following are the required conditions, as stated in Theorem 3 of Halbert White's seminal paper [164]. The data $(X_1, Y_1), \dots, (X_n, Y_n)$ is generated as $X_i = h(Z_i)$, $Y_i = g(Z_i) + \epsilon_i$, where (Z_i, ϵ_i) are mean-zero i.i.d. random draws from some distribution such that $\mathbb{E}[Z_i Z_i^\top]$ and $\mathbb{E}[X_i X_i^\top]$ are finite and nonsingular, and $\mathbb{E}[\epsilon_i^2]$, $\mathbb{E}[Y_i^2 X_i X_i^\top]$, and $\mathbb{E}[X_{ij}^2 X_i X_i^\top]$ are all finite. In addition, we assume that h and g are measurable. Under these conditions,

$$\sqrt{n}(\hat{\theta}_{\text{OLS}} - \theta^*) \Rightarrow \mathcal{N}(0, \Sigma^{-1} V \Sigma^{-1}),$$

where $\theta^* = \arg \min_\theta \mathbb{E}[(Y_i - X_i^\top \theta)^2]$, $\hat{\theta}_{\text{OLS}} = \arg \min_\theta \frac{1}{n} \sum_{i=1}^n (Y_i - X_i^\top \theta)^2$, $\Sigma = \mathbb{E}[X_i X_i^\top]$, $V = \mathbb{E}[(Y_i - X_i^\top \theta^*)^2 X_i X_i^\top]$. Moreover, $\frac{1}{n} X^\top X \rightarrow \Sigma$ and $\frac{1}{n} \sum_{i=1}^n (Y_i - X_i^\top \hat{\theta}_{\text{OLS}})^2 X_i X_i^\top \rightarrow V$ almost surely.

4.5.9 Proofs

4.5.10 Proof of Theorem 15

We show that $\theta^* \in \mathcal{C}_\alpha^{\text{PP}}$ with probability at least $1 - \alpha$; that is, with probability at least $1 - \alpha$ it holds that

$$0 \in \mathcal{R}_\delta(\theta^*) + \mathcal{T}_{\alpha-\delta}(\theta^*).$$

Consider the event $E = \{\Delta_{\theta^*} \in \mathcal{R}_\delta(\theta^*)\} \cap \{\mathbb{E}[g_{\theta^*}(X_i, f(X_i))] \in \mathcal{T}_{\alpha-\delta}(\theta^*)\}$. By a union bound, $P(E) \geq 1 - \alpha$. On the event E , we have that

$$\begin{aligned} \mathbb{E}[g_{\theta^*}(X_i, Y_i)] &= \mathbb{E}[g_{\theta^*}(X_i, Y_i)] - \mathbb{E}[g_{\theta^*}(X_i, f(X_i))] + \mathbb{E}[g_{\theta^*}(X_i, f(X_i))] \\ &= \Delta_{\theta^*} + \mathbb{E}[g_{\theta^*}(X_i, f(X_i))] \in \mathcal{R}_\delta(\theta^*) + \mathcal{T}_{\alpha-\delta}(\theta^*). \end{aligned}$$

The theorem finally follows by invoking the nondegeneracy condition, which ensures that $\mathbb{E}[g_{\theta^*}(X_i, Y_i)] = 0$, so we have shown $0 \in \mathcal{R}_\delta(\theta^*) + \mathcal{T}_{\alpha-\delta}(\theta^*)$.

4.5.11 Proof of Theorem 33

We show that $\theta^* \notin \mathcal{C}_\alpha^{\text{PP}}$ with probability at most α in the limit; that is,

$$\limsup_{n, N \rightarrow \infty} P \left(|\hat{\Delta}_{\theta^*, j} + \hat{g}_{\theta^*, j}^f| > z_{1-\alpha/(2p)} \sqrt{\frac{\hat{\sigma}_{\Delta, j}^2(\theta^*)}{n} + \frac{\hat{\sigma}_{g, j}^2(\theta^*)}{N}}, \forall j \in [p] \right) \leq \alpha.$$

For each $j \in [p]$, the central limit theorem implies that

$$\sqrt{n}(\hat{\Delta}_{\theta^*,j} - \mathbb{E}[\hat{\Delta}_{\theta^*,j}]) \Rightarrow \mathcal{N}(0, \sigma_{\Delta,j}^2(\theta^*)); \quad \sqrt{N}(\hat{g}_{\theta^*,j}^f - \mathbb{E}[\hat{g}_{\theta^*,j}^f]) \Rightarrow \mathcal{N}(0, \sigma_{g,j}^2(\theta^*)),$$

where $\sigma_{\Delta,j}^2(\theta^*)$ is the variance of $g_{\theta^*,j}(X_i, Y_i) - g_{\theta^*,j}(X_i, f(X_i))$ and $\sigma_{g,j}^2(\theta^*)$ is the variance of $g_{\theta^*,j}(X_i, f(X_i))$. Therefore, by Slutsky's theorem, we get

$$\begin{aligned} \sqrt{N}(\hat{\Delta}_{\theta^*,j} + \hat{g}_{\theta^*,j}^f - \mathbb{E}[\hat{\Delta}_{\theta^*,j} + \hat{g}_{\theta^*,j}^f]) &= \sqrt{n}(\hat{\Delta}_{\theta^*,j} - \mathbb{E}[\hat{\Delta}_{\theta^*,j}])\sqrt{\frac{N}{n}} + \sqrt{N}(\hat{g}_{\theta^*,j}^f - \mathbb{E}[\hat{g}_{\theta^*,j}^f]) \\ &\Rightarrow \mathcal{N}\left(0, \frac{1}{p}\sigma_{\Delta,j}^2(\theta^*) + \sigma_{g,j}^2(\theta^*)\right). \end{aligned}$$

This in turn implies

$$\limsup_{n,N \rightarrow \infty} P\left(|\hat{\Delta}_{\theta^*,j} + \hat{g}_j^f(\theta^*) - \mathbb{E}[\hat{\Delta}_{\theta^*,j} + \hat{g}_j^f(\theta^*)]| > z_{1-\alpha/(2p)} \frac{\hat{\sigma}_j}{\sqrt{N}}\right) \leq \frac{\alpha}{p}, \quad (4.13)$$

where $\hat{\sigma}_j^2$ is a consistent estimate of the variance $\frac{1}{p}\sigma_{\Delta,j}^2(\theta^*) + \sigma_{g,j}^2(\theta^*)$. We take $\hat{\sigma}_j^2 = \hat{\sigma}_{\Delta,j}^2(\theta^*)\frac{N}{n} + \hat{\sigma}_{g,j}^2(\theta^*)$; this estimate is consistent since the two terms are individually consistent estimates of the respective variances. Now notice that

$$\mathbb{E}[\hat{\Delta}_{\theta^*} + \hat{g}_{\theta^*}^f] = \mathbb{E}[g_{\theta^*}(X_i, Y_i) - g_{\theta^*}(X_i, f(X_i)) + g_{\theta^*}(\tilde{X}_i, f(\tilde{X}_i))] = \mathbb{E}[g_{\theta^*}(X_i, Y_i)] = 0, \quad (4.14)$$

where the last step follows by the nondegeneracy condition. Putting together (4.13), (4.14), and the choice of $\hat{\sigma}_j$ derived above, and applying a union bound, we get

$$\begin{aligned} \limsup_{n,N \rightarrow \infty} P\left(\exists j \in [p] : |\hat{\Delta}_{\theta^*,j} + \hat{g}_j^f(\theta^*)| > z_{1-\alpha/(2p)} \sqrt{\frac{\hat{\sigma}_{\Delta,j}^2(\theta^*)}{n} + \frac{\hat{\sigma}_{g,j}^2(\theta^*)}{N}}\right) \\ \leq \sum_{j=1}^p \limsup_{n,N \rightarrow \infty} P\left(|\hat{\Delta}_{\theta^*,j} + \hat{g}_j^f(\theta^*)| > z_{1-\alpha/(2p)} \sqrt{\frac{\hat{\sigma}_{\Delta,j}^2(\theta^*)}{n} + \frac{\hat{\sigma}_{g,j}^2(\theta^*)}{N}}\right) \\ = \sum_{j=1}^p \limsup_{n,N \rightarrow \infty} P\left(|\hat{\Delta}_{\theta^*,j} + \hat{g}_j^f(\theta^*) - \mathbb{E}[\hat{\Delta}_{\theta^*,j} + \hat{g}_j^f(\theta^*)]| > z_{1-\alpha/(2p)} \hat{\sigma}_j\right) \\ \leq \sum_{j=1}^p \frac{\alpha}{p} \\ = \alpha. \end{aligned}$$

4.5.12 Proof of Proposition 16

We show that the prediction-powered confidence set constructed in Algorithm 1 is a special case of the prediction-powered confidence set constructed in Theorem 33. The proof then follows directly by the guarantee of Theorem 33.

Since $g_\theta(y) = \theta - y$, we have

$$\hat{\Delta}_\theta \equiv \hat{\Delta} = \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i); \quad \hat{g}_\theta^f = \theta - \frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i).$$

Therefore, the set $\mathcal{C}_\alpha^{\text{PP}}$ from Theorem 33 can be written as

$$\begin{aligned} \mathcal{C}_\alpha^{\text{PP}} &= \left\{ \theta : \left| \theta - \frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i) + \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i) \right| \leq w_\alpha(\theta) \right\} \\ &= \left(\frac{1}{N} \sum_{i=1}^N f(\tilde{X}_i) - \frac{1}{n} \sum_{i=1}^n (f(X_i) - Y_i) \pm w_\alpha(\theta) \right). \end{aligned}$$

This is exactly the set constructed in Algorithm 1, which completes the proof.

4.5.13 Proof of Proposition 17

Like in the proof of Proposition 16, we proceed by showing that the prediction-powered confidence set constructed in Algorithm 2 is a special case of the prediction-powered confidence set constructed in Theorem 33. Then, we simply invoke Theorem 33.

Since $g_\theta(y) = -q + \mathbb{1}\{y \leq \theta\}$, we have

$$\hat{\Delta}_\theta = \frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\}); \quad \hat{g}_\theta^f = -q + \hat{F}(\theta),$$

where $\hat{F}(\theta) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{f(\tilde{X}_i) \leq \theta\}$. Therefore, the set $\mathcal{C}_\alpha^{\text{PP}}$ from Theorem 33 can be written as

$$\begin{aligned} \mathcal{C}_\alpha^{\text{PP}} &= \left\{ \theta : \left| \frac{1}{n} \sum_{i=1}^n (\mathbb{1}\{Y_i \leq \theta\} - \mathbb{1}\{f(X_i) \leq \theta\}) - q + \hat{F}(\theta) \right| \leq w_\alpha(\theta) \right\} \\ &= \left\{ \theta : |\hat{F}(\theta) + \hat{\Delta}_\theta - q| \leq w_\alpha(\theta) \right\}. \end{aligned}$$

This is exactly the set constructed in Algorithm 2. Therefore, the guarantee of Proposition 17 follows by the guarantee of Theorem 33.

4.5.14 Proof of Proposition 18

The proof follows a similar pattern as the previous two propositions, by arguing that the prediction-powered confidence set constructed in Algorithm 3 is a special case of the prediction-powered confidence set constructed in Theorem 33.

Since $g_\theta(x, y) = x(\mu_\theta(x) - y)$, we have

$$\hat{\Delta}_\theta \equiv \hat{\Delta} = \frac{1}{n} \sum_{i=1}^n X_i (f(X_i) - Y_i); \quad \hat{g}_\theta^f = \frac{1}{N} \sum_{i=1}^N \tilde{X}_i (\mu_\theta(\tilde{X}_i) - f(\tilde{X}_i)).$$

These quantities are explicitly computed in Algorithm 3. Moreover, the set $\mathcal{C}_\alpha^{\text{PP}}$ constructed in Algorithm 3 exactly follows the recipe of Theorem 33, so the proof immediately follows.

4.5.15 Proof of Proposition 19

For linear regression, we can derive more powerful prediction-powered confidence intervals than those implied by Theorem 15 by exploiting the linearity of the least-squares estimator.

Recall that Theorem 33 assumes that $\frac{n}{N} \rightarrow p$, for some fraction $p \in (0, 1)$.

Theorem 3 of White [163] implies that

$$\sqrt{n}(\hat{\Delta} - \Delta) \Rightarrow \mathcal{N}(0, W); \quad \sqrt{N}(\tilde{\theta}^f - \theta^f) \Rightarrow \mathcal{N}(0, W'),$$

for appropriately defined covariance matrices W and W' , where $\theta^f = (\mathbb{E}[X_i X_i^\top])^{-1} \mathbb{E}[X_i f(X_i)]$ and $\Delta = (\mathbb{E}[X_i X_i^\top])^{-1} \mathbb{E}[X_i (f(X_i) - Y_i)]$. With this, we can write the target estimand as $\theta^* = (\mathbb{E}[X_i X_i^\top])^{-1} \mathbb{E}[X_i Y_i] = \theta^f - \Delta$.

Combining Theorem 3 of White with Slutsky's theorem, we get

$$\sqrt{N}(\hat{\theta}^{\text{PP}} - \theta^*) = \sqrt{N}(\tilde{\theta}^f - \theta^f) - \sqrt{n}(\hat{\Delta} - \Delta) \sqrt{\frac{N}{n}} \Rightarrow \mathcal{N}\left(0, W \frac{1}{p} + W'\right).$$

White also shows that V and \tilde{V} , as defined in Algorithm 4, are consistent estimates of W and W' , respectively. Therefore, $\hat{\theta}^{\text{PP}}$ is asymptotically normal and consistent, and we have a consistent estimate of its covariance. In particular,

$$V_{j^*j^*} \frac{N}{n} + \tilde{V}_{j^*j^*} \rightarrow W_{j^*j^*} \frac{1}{p} + W'_{j^*j^*}.$$

This means that we can construct asymptotically valid confidence intervals via a normal approximation by choosing width $z_{1-\alpha/2} \sqrt{V_{j^*j^*} \frac{N}{n} + \tilde{V}_{j^*j^*}} \sqrt{\frac{1}{N}} = z_{1-\alpha/2} \sqrt{\frac{V_{j^*j^*}}{n} + \frac{\tilde{V}_{j^*j^*}}{N}}$, and this is precisely what Algorithm 4 accomplishes.

4.5.16 Proof of Theorem 20

Define

$$L(\theta) = \mathbb{E}[\ell_\theta(X_i, Y_i)], \quad L^f(\theta) = \mathbb{E}[\ell_\theta(X_i, f(X_i))].$$

By the definition of θ^* , we have

$$\begin{aligned} \tilde{L}^f(\theta^*) &= (\tilde{L}^f(\theta^*) - L(\theta^*)) + (L(\theta^*) - L(\tilde{\theta}^f)) + (L(\tilde{\theta}^f) - \tilde{L}^f(\tilde{\theta}^f)) + \tilde{L}^f(\tilde{\theta}^f) \\ &\leq (\tilde{L}^f(\theta^*) - L(\theta^*)) + (L(\tilde{\theta}^f) - \tilde{L}^f(\tilde{\theta}^f)) + \tilde{L}^f(\tilde{\theta}^f). \end{aligned}$$

By applying the validity of the confidence bounds, a union bound implies that with probability $1 - \alpha$ we have

$$\begin{aligned} \tilde{L}^f(\theta^*) &\leq (L^f(\theta^*) - L(\theta^*)) + (L(\tilde{\theta}^f) - L^f(\tilde{\theta}^f)) + \tilde{L}^f(\tilde{\theta}^f) + \mathcal{T}_{\frac{\alpha-\delta}{2}}^u(\theta^*) - \mathcal{T}_{\frac{\alpha-\delta}{2}}^l(\tilde{\theta}^f) \\ &= -\Delta_{\theta^*} + \Delta_{\tilde{\theta}^f} + \tilde{L}^f(\tilde{\theta}^f) + \mathcal{T}_{\frac{\alpha-\delta}{2}}^u(\theta^*) - \mathcal{T}_{\frac{\alpha-\delta}{2}}^l(\tilde{\theta}^f) \\ &\leq -\mathcal{R}_{\delta/2}^l(\theta^*) + \mathcal{R}_{\delta/2}^u(\tilde{\theta}^f) + \tilde{L}^f(\tilde{\theta}^f) + \mathcal{T}_{\frac{\alpha-\delta}{2}}^u(\theta^*) - \mathcal{T}_{\frac{\alpha-\delta}{2}}^l(\tilde{\theta}^f). \end{aligned}$$

Therefore, with probability $1 - \alpha$ we have that $\theta^* \in \mathcal{C}_\alpha^{\text{PP}}$, as desired.

4.5.17 Proof of Theorem 22

Notice that we can write $\mathbb{E}_{\mathbb{Q}_Y}[\nu(Y)] = \nu^\top \mathbb{Q}_Y$, where on the right-hand side we are treating $\nu = (\nu(1), \dots, \nu(K))$ and $\mathbb{Q}_Y = (\mathbb{Q}_Y(1), \dots, \mathbb{Q}_Y(K))$ as vectors of length K . We can write similar expressions for $\mathbb{Q}_f, \widehat{\mathbb{Q}}_Y$, etc. Using this notation, by triangle inequality we have

$$|\theta^* - \nu^\top \widehat{\mathbb{Q}}_Y| = |\nu^\top \mathbb{Q}_Y - \nu^\top \widehat{\mathbb{Q}}_Y| \leq |\nu^\top \widehat{\mathcal{K}}^{-1}(\mathbb{Q}_f - \widehat{\mathbb{Q}}_f)| + |\nu^\top \mathcal{K}^{-1} \mathbb{Q}_f - \nu^\top \widehat{\mathcal{K}}^{-1} \mathbb{Q}_f|. \quad (4.15)$$

We bound the first term using Hölder's inequality,

$$|\nu^\top \widehat{\mathcal{K}}^{-1}(\mathbb{Q}_f - \widehat{\mathbb{Q}}_f)| \leq \|\nu^\top \widehat{\mathcal{K}}^{-1}\|_1 \|\mathbb{Q}_f - \widehat{\mathbb{Q}}_f\|_\infty.$$

For the second term, we write

$$|\nu^\top \mathcal{K}^{-1} \mathbb{Q}_f - \nu^\top \widehat{\mathcal{K}}^{-1} \mathbb{Q}_f| = |\nu^\top \widehat{\mathcal{K}}^{-1}(\widehat{\mathcal{K}} - \mathcal{K})\mathcal{K}^{-1} \mathbb{Q}_f|.$$

In the above equation, the factor on the right, $\mathcal{K}^{-1} \mathbb{Q}_f$, is exactly equal to \mathbb{Q}_Y , and thus lives on the simplex, which we denote by Δ . Using this fact and Hölder's inequality,

$$|\nu^\top \widehat{\mathcal{K}}^{-1}(\widehat{\mathcal{K}} - \mathcal{K})\mathcal{K}^{-1} \mathbb{Q}_f| \leq \sup_{q \in \Delta} |\nu^\top \widehat{\mathcal{K}}^{-1}(\widehat{\mathcal{K}} - \mathcal{K})q| \leq \|\nu^\top \widehat{\mathcal{K}}^{-1}\|_1 \sup_{q \in \Delta} \|(\widehat{\mathcal{K}} - \mathcal{K})q\|_\infty.$$

Next, we have

$$\sup_{q \in \Delta} \|(\widehat{\mathcal{K}} - \mathcal{K})q\|_\infty = \max_{k \in [K]} \|\widehat{\mathcal{K}}_k - \mathcal{K}_k\|_\infty,$$

where \mathcal{K}_k indexes the k -th column of \mathcal{K} . This yields the expression

$$\|\nu^\top \widehat{\mathcal{K}}^{-1}\|_1 \sup_{q \in \Delta} \|(\widehat{\mathcal{K}} - \mathcal{K})q\|_\infty = \|\nu^\top \widehat{\mathcal{K}}^{-1}\|_1 \max_{k \in [K]} \|\widehat{\mathcal{K}}_k - \mathcal{K}_k\|_\infty.$$

Putting everything together and going back to (4.15), we have

$$|\nu^\top \mathbb{Q}_Y - \nu^\top \widehat{\mathbb{Q}}_Y| \leq \|\nu^\top \widehat{\mathcal{K}}^{-1}\|_1 \left(\|\mathbb{Q}_f - \widehat{\mathbb{Q}}_f\|_\infty + \max_{k \in [K]} \|\widehat{\mathcal{K}}_k - \mathcal{K}_k\|_\infty \right). \quad (4.16)$$

Since $\|\nu^\top \widehat{\mathcal{K}}^{-1}\|_1$ can be evaluated empirically, it remains to bound the distributional distances $\|\mathbb{Q}_f - \widehat{\mathbb{Q}}_f\|_\infty$ and $\max_{k \in [K]} \|\widehat{\mathcal{K}}_k - \mathcal{K}_k\|_\infty$.

For the first term, we can simply apply the DKWM inequality [60, 113], which gives

$$\|\mathbb{Q}_f - \widehat{\mathbb{Q}}_f\|_\infty \leq \sqrt{\frac{2}{N} \log \frac{2}{\alpha - \delta}} \quad (4.17)$$

with probability $1 - (\alpha - \delta)$. See [36] for details.

For the second term, $\max_{k \in [K]} \|\widehat{\mathcal{K}}_k - \mathcal{K}_k\|_\infty$, since we only have n observations for estimation, we use a more adaptive concentration result. In particular, for each $l, k \in [K]$, $n(k)\widehat{\mathcal{K}}_{l,k}$ (conditional on the k -th column) follows a binomial distribution with $n(k)$ draws and success probability $\mathcal{K}_{l,k}$. Therefore, if we let

$$C_{l,k} = \left\{ p : n(k)\widehat{\mathcal{K}}_{l,k} \in \left(F_{\text{Binom}(n(k),p)}^{-1} \left(\frac{\delta}{2K^2} \right), F_{\text{Binom}(n(k),p)}^{-1} \left(1 - \frac{\delta}{2K^2} \right) \right) \right\},$$

where $F_{\text{Binom}(n(k),p)}$ denotes the Binomial CDF, then by a union bound:

$$P \left(\max_{k \in [K]} \|\widehat{\mathcal{K}}_k - \mathcal{K}_k\|_\infty \geq \max_{l,k \in [K]} \max_{p \in C_{l,k}} |\widehat{\mathcal{K}}_{l,k} - p| \right) \leq \delta. \quad (4.18)$$

Combining equations (4.16), (4.17) and (4.18) yields the final result.

4.5.18 Proof of Corollary 34

The proof follows by instantiating the terms in Theorem 15. In particular, we have $\mathbb{E}[g_\theta(f(X_i))] = \theta - \mathbb{E}[f(X_i)]$, hence it is valid to construct $\mathcal{T}_{\alpha-\delta}(\theta)$ as:

$$\mathbb{E}[g_\theta(f(X_i))] \in \mathcal{T}_{\alpha-\delta}(\theta) = \theta - (f_{\alpha-\delta}^l, f_{\alpha-\delta}^u).$$

Therefore, the condition $0 \in \mathcal{R}_\delta + \mathcal{T}_{\alpha-\delta}(\theta)$ becomes

$$0 \in (\mathcal{R}_\delta^l, \mathcal{R}_\delta^u) + \theta - (f_{\alpha-\delta}^l, f_{\alpha-\delta}^u),$$

which after rearranging and simplifying is equivalent to

$$\theta \in (f_{\alpha-\delta}^l - \mathcal{R}_\delta^u, f_{\alpha-\delta}^u - \mathcal{R}_\delta^l).$$

This set exactly matches the set $\mathcal{C}_\alpha^{\text{PP}}$ constructed in Algorithm 10.

4.5.19 Proof of Corollary 35

The proof follows by instantiating the terms in Theorem 15. First, we have $\mathbb{E}[g_\theta(f(X_i))] = -q + P(f(X_i) \leq \theta)$; therefore, it is valid to construct $\mathcal{T}_{\alpha-\delta}(\theta)$ as:

$$\mathbb{E}[g_\theta(f(X_i))] \in \mathcal{T}_{\alpha-\delta}(\theta) = -q + (\widehat{F}_{\alpha-\delta}^l(\theta), \widehat{F}_{\alpha-\delta}^u(\theta)).$$

Therefore, the condition $0 \in \mathcal{R}_\delta(\theta) + \mathcal{T}_{\alpha-\delta}(\theta)$ becomes

$$q \in (\widehat{F}_{\alpha-\delta}^l(\theta) + \mathcal{R}_\delta^l(\theta), \widehat{F}_{\alpha-\delta}^u(\theta) + \mathcal{R}_\delta^u(\theta)),$$

which matches the condition used to form $\mathcal{C}_\alpha^{\text{PP}}$ in Algorithm 11.

4.5.20 Proof of Corollary 36

We instantiate the relevant terms in Theorem 15. Here, we have that $\mathbb{E}[g_\theta(X_i, f(X_i))] = \mathbb{E}\left[-X_i f(X_i) + X_i \frac{1}{1+\exp(-X_i^\top \theta)}\right]$. Note that, because X_i is coordinatewise bounded, and because $Y_i, \frac{1}{1+\exp(-X_i^\top \theta)} \in [0, 1]$, we have $|(g_\theta(X_i, f(X_i)))_j| \leq B_j$ almost surely. Therefore, we can construct $\mathcal{T}_{\alpha-\delta}(\theta)$ as:

$$\begin{aligned} \mathbb{E}[g_\theta(X_i, f(X_i))] \in \mathcal{T}_{\alpha-\delta}(\theta) &= (g_{\alpha-\delta}^l(\theta), g_{\alpha-\delta}^u(\theta)) \\ &= (g_{\alpha-\delta,1}^l(\theta), g_{\alpha-\delta,1}^u(\theta)) \times \cdots \times (g_{\alpha-\delta,d}^l(\theta), g_{\alpha-\delta,d}^u(\theta)). \end{aligned}$$

Since the rectifier has no dependence on θ , the condition $0 \in \mathcal{R}_\delta(\theta) + \mathcal{T}_{\alpha-\delta}(\theta)$ becomes

$$0 \in (\mathcal{R}_{\delta,j}^l, \mathcal{R}_{\delta,j}^u) + (g_{\alpha-\delta,j}^l(\theta), g_{\alpha-\delta,j}^u(\theta)), \quad \forall j \in [d],$$

which matches the condition in $\mathcal{C}_\alpha^{\text{PP}}$ in Algorithm 12.

4.5.21 Confidence intervals for the mean

We give an overview of off-the-shelf confidence intervals for the mean. We state the results for two observation models: first for the i.i.d. sampling model considered in the main body and then for the finite-population setting discussed in Appendix 4.5.2. In both cases, we provide a construction with nonasymptotic guarantees and one with asymptotic guarantees.

For the nonasymptotic confidence intervals, we rely on the results of Waudby-Smith and Ramdas [161], specifically their Theorem 3 and Theorem 4. We opt for these results because of their strong practical performance, which is primarily driven by variance adaptivity. These results assume that the observed random variables are bounded within a known interval. Without loss of generality we assume that the observations are bounded in $[0, 1]$ (otherwise we can always normalize the observations to $[0, 1]$).

For the asymptotic confidence intervals, we rely on the central limit theorem (CLT) and its variant for sampling without replacement; see [62, 81] for classical references.

4.5.22 Inference with i.i.d. samples

In the following two results, assume that we observe $Z_1, \dots, Z_n \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}$ and let $\mu = \mathbb{E}[Z_i]$.

Proposition 37 (Nonasymptotic CI: Theorem 3 in [161]). *Assume $\text{supp}(\mathbb{P}) \subseteq [0, 1]$. Let*

$$\hat{\mu}_t = \frac{0.5 + \sum_{j=1}^t Z_j}{t+1}, \quad \hat{\sigma}_t^2 = \frac{0.25 + \sum_{j=1}^t (Z_j - \hat{\mu}_t)^2}{t+1}, \quad \lambda_t = \sqrt{\frac{2 \log(2/\alpha)}{n \hat{\sigma}_{t-1}^2}}.$$

For every $m \in [0, 1]$, define the supermartingale:

$$M_t(m) = \frac{1}{2} \max \left\{ \prod_{j=1}^t \left(1 + \min \left(\lambda_j, \frac{0.5}{m} \right) (Z_j - m) \right), \prod_{j=1}^t \left(1 - \min \left(\lambda_j, \frac{0.5}{1-m} \right) (Z_j - m) \right) \right\}.$$

Let

$$\mathcal{C} = \bigcap_{t=1}^n \{m \in [0, 1] : M_t(m) < 1/\alpha\}.$$

Then,

$$P(\mu \in \mathcal{C}) \geq 1 - \alpha.$$

Intuitively, the supermartingale $M_t(m)$ should be thought of as the amount of evidence against m being the true mean. That is, $M_t(m)$ being big suggests that m is unlikely to be the true mean, so the final confidence set is the collection of all m for which the amount of such evidence is small.

For large n , computing the intersection in the definition of \mathcal{C} can be intractable, so we conservatively choose a subsequence of $1, \dots, n$ for the computation.

Proposition 38 (Asymptotic CI: CLT interval). *Assume \mathbb{P} has a finite second moment. Let*

$$\mathcal{C} = \left(\frac{1}{n} \sum_{i=1}^n Z_i \pm z_{1-\alpha/2} \frac{\hat{\sigma}}{\sqrt{n}} \right),$$

where $\hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Z_i - \frac{1}{n} \sum_{j=1}^n Z_j)^2}$. Then,

$$\liminf_{n \rightarrow \infty} P(\mu \in \mathcal{C}) \geq 1 - \alpha.$$

4.5.23 Inference on a finite population

In the following two results, we assume that there exists a *fixed* sequence Z_1, \dots, Z_N , and we observe $\{Z_i : i \in \mathcal{I}\}$, where $\mathcal{I} = \{i_1, \dots, i_n\}$ is a uniform random subset of $[N]$ with cardinality n . We let $\mu = \frac{1}{N} \sum_{i=1}^N Z_i$. For the asymptotic result, we assume that Z_1, \dots, Z_N is the first N entries of an infinite underlying sequence Z_1, Z_2, \dots .

Proposition 39 (Nonasymptotic CI: Theorem 4 in [161]). *Assume $Z_i \in [0, 1]$, $i \in [N]$. Let*

$$\hat{\mu}_t = \frac{0.5 + \sum_{j=1}^t Z_{i_j}}{t+1}, \quad \hat{\sigma}_t^2 = \frac{0.25 + \sum_{j=1}^t (Z_{i_j} - \hat{\mu}_t)^2}{t+1}, \quad \lambda_t = \sqrt{\frac{2 \log(2/\alpha)}{n \hat{\sigma}_{t-1}^2}}.$$

For every $m \in [0, 1]$, define the supermartingale:

$$M_t(m) = \frac{1}{2} \max \left\{ \prod_{j=1}^t \left(1 + \min \left(\lambda_j, \frac{0.5}{\mu_t(m)} \right) (Z_{i_j} - \mu_t(m)) \right), \right. \\ \left. \prod_{j=1}^t \left(1 - \min \left(\lambda_j, \frac{0.5}{1 - \mu_t(m)} \right) (Z_{i_j} - \mu_t(m)) \right) \right\},$$

where $\mu_t(m) = \frac{Nm - \sum_{j=1}^{t-1} Z_{i_j}}{N-t+1}$ is the putative mean. Let

$$\mathcal{C} = \bigcap_{t=1}^n \{m \in [0, 1] : M_t(m) < 1/\alpha\}.$$

Then,

$$P(\mu \in \mathcal{C}) \geq 1 - \alpha.$$

Proposition 40 (Asymptotic CI: CLT for sampling without replacement). *Let $\sigma^2 = \frac{1}{N} \sum_{i=1}^N (Z_i - \mu)^2$, and $\hat{\sigma}^2 = \frac{1}{n} \sum_{i \in \mathcal{I}} (Z_i - \hat{\mu})^2$. Assume that μ and σ have a limit and that $n/N \rightarrow p$ for some $p \in (0, 1)$. Let*

$$\mathcal{C} = \left(\frac{1}{n} \sum_{i \in \mathcal{I}} Z_i \pm z_{1-\alpha/2} \frac{\hat{\sigma}}{\sqrt{n}} \sqrt{\frac{N-n}{N}} \right).$$

Then,

$$\liminf_{n, N \rightarrow \infty} P(\mu \in \mathcal{C}) \geq 1 - \alpha.$$

4.5.24 Comparison to baseline procedures

We compare prediction-powered inference with two baseline procedures that also combine labeled and unlabeled data in performing statistical inference. The baselines are:

1. **Post-prediction inference.** We use the post-prediction inference procedure of Wang et al. [159] to estimate ordinary least-squares (OLS) coefficients. The procedure first fits a regression r to predict Y_i from $f(X_i)$ on the gold-standard dataset. Subsequently, the regression function is used to correct the imputed labels on the unlabeled dataset. Confidence intervals are formed using the $r(f(\tilde{X}_i))$ as if they were gold-standard data. This procedure has no theoretical guarantees in general and requires strong distributional assumptions on the relationship between Y_i and $f(X_i)$ to provide coverage. Our experiments indicate that this approach fails to cover in realistic conditions.
2. **Semi-supervised mean estimation.** The semi-supervised mean estimation procedure of Zhang and Bradic [175] involves cross-fitting a (possibly-regularized) linear model on K distinct folds of the gold-standard dataset. The average of the K model predictions on each unlabeled data point is taken as its corresponding prediction \hat{Y} , and the average bias $\hat{Y} - Y$ of the K models is also computed and used to debias the resulting mean estimate. The formal validity of this approach applies to mean estimation and requires the cross-fitting of linear models; it does not have formal guarantees for more flexible model classes. For this reason, it provides little improvement over the classical confidence interval in our experiments, since the variance reduction possible with linear models is typically limited.

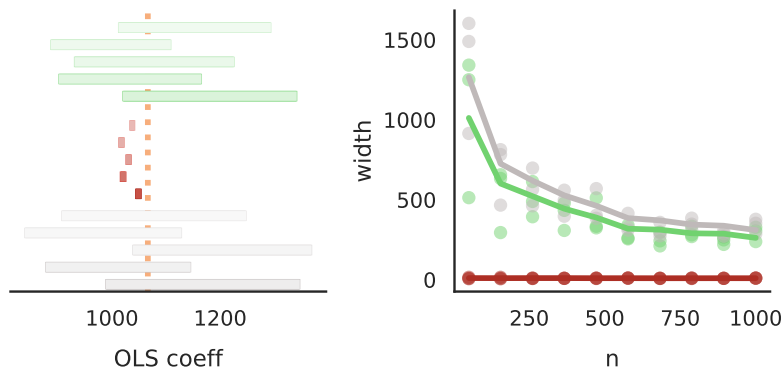


Figure 4.2: **Comparison to the post-prediction inference procedure.** On the left are five independent random draws of intervals with $n = 1000$. On the right is a line plot of interval width as a function of n , averaged over 100 independent trials. Five draws of interval widths are shown as a scatter plot at their respective n . The post-prediction inference approach is shown in red, the classical approach is in gray, and the prediction-powered approach is in green. The post-prediction inference approach has diminishing coverage in the experiment.

4.5.25 Experimental protocol

We evaluate the methods on an income prediction task on the same census dataset used for the logistic regression experiments in the main text. In the case of the semi-supervised baseline, the goal is to estimate the mean income in California in the year 2019 among employed individuals using a small amount of labeled data and a large amount of covariates. In the case of the post-prediction inference baseline, the target of inference is the OLS coefficient between age and income. The setup is the same as the logistic regression experiment described in the main text (including the use of the Folktables [57] interface and the gradient-boosted tree [46] as the predictor).

4.5.26 Comparison to post-prediction inference

Results of the post-prediction inference protocol as compared to the classical and prediction-powered approaches are shown in Figure 4.2 for the previously-described OLS coefficient between age and income. The procedure does not cover at the proper rate and the intervals are biased.

4.5.27 Comparison to semi-supervised mean estimation

Results of the semi-supervised mean estimation protocol as compared to the classical and prediction-powered approaches are shown in Figure 4.3 for the previously described mean

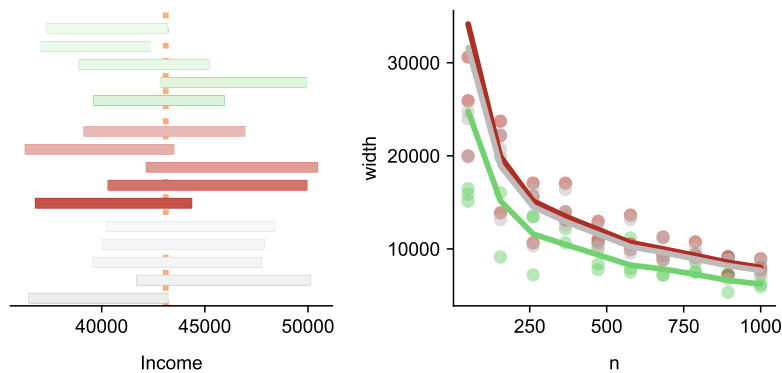


Figure 4.3: **Comparison to the semi-supervised mean estimation procedure.** The plot is the same as in Figure 4.2, but with semi-supervised inference shown in red. The semi-supervised intervals have a similar width to the classical ones in this experiment, while the prediction-powered intervals dominates.

income estimation task. The prediction-powered intervals dominate both the semi-supervised intervals and the classical ones in the experiment for all values of n .

4.5.28 Cases where prediction-powered inference is underpowered

Since standard confidence intervals scale with the standard error of the estimator, prediction-powered inference is powerful when a machine-learning model can provide a reduction in the estimator variance. At a high level, this happens when N is large enough relative to n and the model is accurate enough. This was the case in all the experiments shown in the main text. This section precisely quantifies what it means to have an accurate enough model and large enough N . Corroborating the theory, we present two cases where classical inference outperforms prediction-powered inference: one where the model is not good enough and another where N is too small.

4.5.29 Mathematical derivation

Consider the case of mean estimation, $\theta^* = \mathbb{E}[Y_i]$. The widths of the classical confidence interval based on the central limit theorem and the prediction-powered confidence interval based on Algorithm 1 scale with $\text{Var}(\hat{\theta}^{\text{class}})$ and $\text{Var}(\hat{\theta}^{\text{PP}})$, respectively, where $\hat{\theta}^{\text{class}}$ and $\hat{\theta}^{\text{PP}}$ are defined in Section 4.1.3. The classical estimator has variance equal to

$$\text{Var}(\hat{\theta}^{\text{class}}) = \frac{1}{n} \text{Var}(Y_i).$$

The variance of the prediction-powered estimator equals

$$\text{Var}(\hat{\theta}^{\text{PP}}) = \frac{1}{N} \text{Var}(f(X_i)) + \frac{1}{n} \text{Var}(f(X_i) - Y_i).$$

Therefore, the prediction-powered confidence interval will be tighter when

$$\frac{1}{N} \text{Var}(f(X_i)) + \frac{1}{n} \text{Var}(f(X_i) - Y_i) < \frac{1}{n} \text{Var}(Y_i).$$

Since the predictions $f(X_i)$ will typically have a variance that is of the same order as the variance of Y_i , if $N \approx n$ one should not expect prediction-powered inference to help. Gains are expected when $N \gg n$. In that case, $\frac{1}{N} \text{Var}(f(X_i)) \ll \frac{1}{n} \text{Var}(f(X_i) - Y_i)$, and thus prediction-powered inference helps when

$$\text{Var}(f(X_i) - Y_i) < \text{Var}(Y_i).$$

In other words, prediction-powered inference gives tighter confidence intervals when the predictions explain away some of the outcome variance.

To gain further intuition, suppose that the outcomes are binary, $Y_i \sim \text{Bern}(p)$, where $\text{Bern}(p)$ denotes the Bernoulli distribution with parameter p . In this case, $\theta^* = p$. For simplicity, suppose that $P(f(X_i) = 0|Y_i = 1) = P(f(X_i) = 1|Y_i = 0) = \eta$. Then, a direct variance calculation gives $\text{Var}(f(X_i) - Y_i) = \eta - \eta^2(1 - 2p)^2$ and $\text{Var}(Y_i) = p(1 - p)$. This allows for a direct comparison of the variances in terms of the outcome bias p and model error η . For example, when $p = 0.5$, the model error η has to be smaller than 25% for prediction-powered inference to yield smaller intervals; when $p = 0.1$, meaning the outcomes themselves have low variance, the model error η has to be smaller than about 9.5%. In general, the lower the variance of the outcome, the lower the model error has to be for prediction-powered inference to be helpful.

Putting everything together, the main takeaway is as follows: prediction-powered inference should only be applied when N is (preferably substantially) larger than n , and when the model has a high enough predictive accuracy to explain away some of the outcome variance. While this derivation focused on mean estimation, a similar intuition holds for other estimation problems.

4.5.30 Inaccurate machine-learning model

We repeat the deforestation analysis experiment from the main text. However, instead of a gradient-boosted tree, we use a linear regression model for prediction. This degrades predictive performance enough that the classical baseline outperforms the prediction-powered approach. See Figure 4.4 for the results. Due to the reduction of power, for the same null hypothesis tested in the main text, the prediction-powered approach requires $n = 40$ data points to reject, while the classical baseline requires $n = 35$.

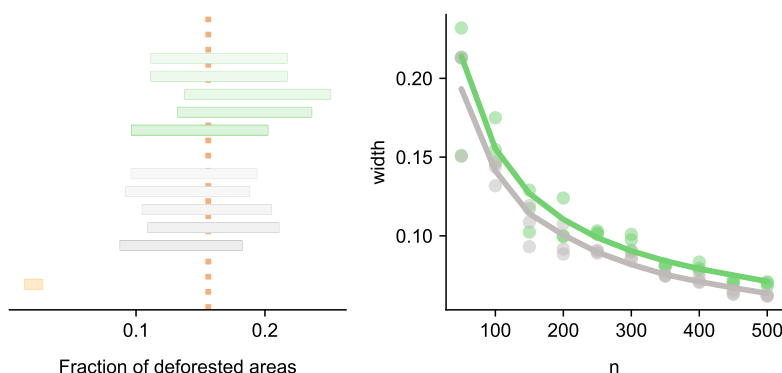


Figure 4.4: **Deforestation analysis with a linear model.** This is the same figure as Figure 4.1D, with the same color coding; the prediction-powered approach is green, the classical approach is gray, and the imputation approach is gold. However, the gradient-boosted tree is replaced with an ordinary linear regression. The drop in performance causes the classical intervals to outperform the prediction-powered intervals in terms of power.

4.5.31 Unlabeled dataset is too small

We repeat the AlphaFold-based proteomic analysis from the main text. However, $N = 1000$ data points are randomly chosen as the unlabeled dataset. The rest of the procedure is performed exactly the same way as described in the main text. The decrease in the unlabeled sample size leads to a reduction in power, and in the regime $n > N$, the classical baseline outperforms the prediction-powered approach. See Figure 4.5 for the results. For the same null hypothesis as in the main text, the prediction-powered approach requires $n = 869$ data points to reject, while the classical baseline requires $n = 652$.

4.5.32 Experimental particulars

4.5.33 Relating protein structure and post-translational modifications

The predictive model of whether a sequence position is in an intrinsically disordered region (IDR), f , is a logistic regression model that maps the relative solvent-accessible surface area (RSA) of each position, computed based on the AlphaFold-predicted structure using Bio.PDB [76], to a probability that the position is in an IDR. Following Bludau et al. [25], the RSA was locally smoothed with a window of 5, 10, 15, 20, 25, 30, or 35 amino acids, and a sigmoid function was used to predict disorder from this smoothed RSA quantity. To fit the sigmoid, we used the data in [25] that had IDR labels but no PTM labels. The smoothing window size used for the final model was the value that resulted in the lowest variance of the bias, $Y - f$, on this data.

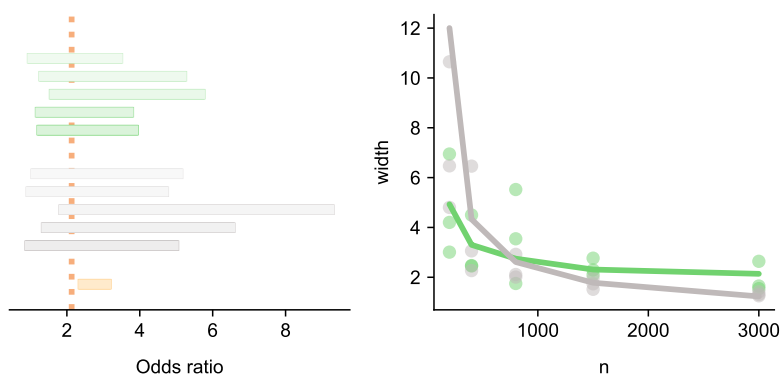


Figure 4.5: **AlphaFold analysis with a small unlabeled dataset.** This is the same figure as Figure 4.1A, with the same color coding; the prediction-powered approach is green, the classical approach is gray, and the imputation approach is gold. However, here N is taken to be 1000. It can be seen that, when $n > N$, the classical baseline outperforms the prediction-powered one.

Figure 4.1A in the main text presents results on estimating the odds ratio between intrinsic disorder and phosphorylation, a common type of post-translational modification (PTM). Figure 4.6 shows analogous results on estimating the odds ratio between intrinsic disorder and two other types of PTMs, ubiquitination and acetylation. The confidence intervals shown in the left panel of Figure 4.6 and Figure 4.1A in the main text were computed with $n = 400$ labeled data points.

4.5.34 Galaxy classification

We fine-tune a ResNet50 [80] on the training split of the Galaxy Zoo 2 data with a batch size of 32 and a learning rate of 0.0001 using Adam [92]. We tune the entire backbone, not just the last layer. We use the remaining validation split as our labeled and unlabeled data, taking $n \in \{50, 100, 200, 300, 500, 750, 1000\}$. We use Algorithm 1 for the prediction-powered approach, and Proposition 38 for the classical and imputation approaches. The confidence intervals shown in the left panel of Figure 4.1B in the main text were computed with $n = 366$ labeled data points.

4.5.35 Distribution of gene expression levels

We used the transformer model developed and trained by Vaishnav et al. [150] to predict gene expression level, with the following modification that we found improved predictive performance. Given n labeled data points, five were randomly selected and used to train an affine (two-parameter) function mapping the scalar prediction of the transformer in [150] to a

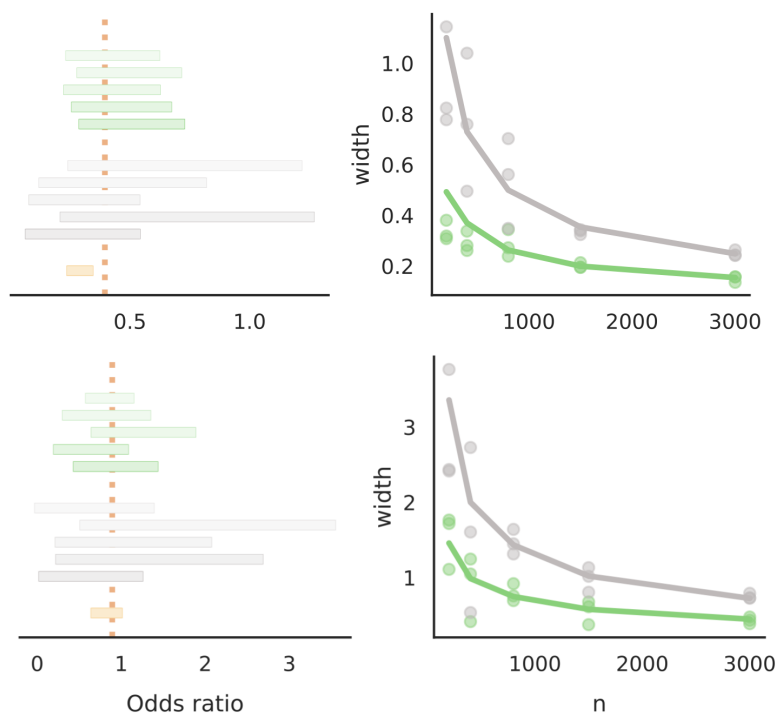


Figure 4.6: **Confidence intervals on odds ratio between intrinsic disorder and two types of post-translational modifications**, ubiquitination (top) and acetylation (bottom). Following Figure 4.1 in the main text, the left panel shows prediction-powered (green) and classical (gray) confidence intervals computed with five random splits of labeled and unlabeled data, as well as the imputation (gold) confidence interval computed using all the unlabeled data. The true value is denoted by the dashed orange line. The right panel shows the average interval width for varying values of n , the number of labeled data points, and the width for five randomly chosen trials.

prediction of the conditional median of the label, using quantile regression. The predictions of this final model were used for the unlabeled dataset, and the remaining $n - 5$ data points that were not used to fine-tune the transformer model were used as the labeled dataset. The confidence intervals shown in the left panel of Figure 4.1C in the main text were computed with $n = 2000$ labeled data points.

We plot results analogous to Figure 4.1C in the main text for the 0.25- and 0.75-quantiles in Figure 4.7.

4.5.36 Estimating deforestation in the Amazon

The machine-learning model given by [140] outputs forest-cover predictions at 30m resolution for 3192 data points. We correspond these by latitude and longitude with gold-standard

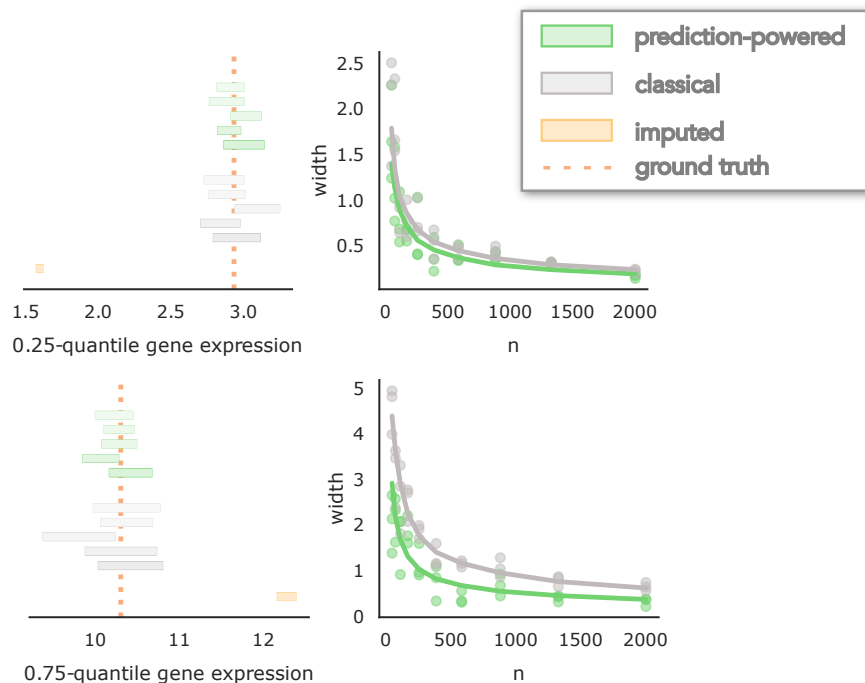


Figure 4.7: **Confidence intervals on gene expression quantiles** for $q = 0.25$ (top) and $q = 0.75$ (bottom). Following Figure 4.1 in the main text, the left panel shows prediction-powered (green) and classical (gray) confidence intervals computed with five random splits of labeled and unlabeled data. The right panel shows the average interval width for varying values of n , the number of labeled data points, as well as the width for five randomly chosen trials.

data points labeled as one of $\{\text{deforestation, no deforestation}\}$ from [32]. In the first step, we split off half of the data to train a histogram-based gradient-boosted tree to predict deforestation labels from the forest-cover predictions. We take a random sample of $n = 100$ data points as the gold-standard data, and try to cover the true fraction of deforestation events on the $N = 1596$ remaining data points. We use Algorithm 1 and Proposition 38 to produce the prediction-powered confidence interval and the classical and imputation intervals, respectively. The confidence intervals shown in the left panel of Figure 4.1D in the main text were computed with $n = 200$ labeled data points.

4.5.37 Relationship between income and private health insurance

We train a gradient-boosted tree [46] on the California Census data from 2018 acquired using the Folktables [57] interface. The tree takes as input several covariates such as income, race, and sex, to predict whether an individual has private health insurance coverage. In the new year, 2019, we use $n \in \{200, 300, 500, 1000, 2000, 5000, 10000\}$ labeled data points. We use

Algorithm 3 to produce the prediction-powered confidence interval and the standard CLT confidence interval for the classical and imputation approaches. The confidence intervals shown in the left panel of Figure 4.1E in the main text were computed with $n = 2000$ labeled data points.

4.5.38 Relationship between age and income

The setting is the same as the above experiment on income and private health insurance, the main difference being that income is used as the target, and not as a covariate. We used Algorithm 4 to produce the prediction-powered confidence interval and the standard CLT confidence interval for the classical and imputation approaches. The confidence intervals shown in the left panel of Figure 4.1F in the main text were computed with $n = 2000$ labeled data points.

4.5.39 Counting plankton

We fine-tune a ResNet152 [80] on the WHOI-Plankton dataset [119] in the years 2006-2013 for two epochs with a batch size of 32 and a learning rate of 0.0001 using AdamW [109], with 5% of the data saved for validation. We tune the entire backbone, not just the last layer. Then we test in the year 2014, using all available data. We use Theorem 22 to produce the prediction-powered intervals and Proposition 38 for the imputation approach. The confidence intervals shown in the left panel of Figure 4.1G in the main text were computed with $n = 89471$ labeled data points.

Chapter 5

Using Synthetic Data for Model Evaluation

5.1 Introduction

Our goal is to evaluate machine learning systems—assessing their accuracy, fairness, and other metrics—with as few data points as possible. This goal is important for reducing the human effort required to collect extensive validation datasets [77] for such tasks. Towards that end, we will explore an approach called *autoevaluation*, wherein we evaluate models in a two-stage procedure: (i). Produce synthetic labels using AI on a large unlabeled dataset, and (ii). evaluate AI models using the synthetic labels.

Autoevaluation can save months or years of time and potentially millions of dollars in annotation costs; see, e.g., Scale AI, or the recent work of Zheng et al. [176] using `gpt-4` to rank alternative language models’ answers to questions with high agreement with human annotators. However, the synthetic labels may not be trustworthy, especially for the purpose of certifying a model’s worst-case safety, multi-group accuracy and fairness, or to understand if observed differences between models are significant. This motivates the need for serious statistical inquiry on the general question of autoevaluation.

This chapter introduces methods for *autoevaluation done right*. Given a small amount of human data and a large amount of synthetic data, we will construct autoevaluation procedures that combine these datasets to get better estimates of performance.

In other words, our methods will increase the effective sample size of human data with-

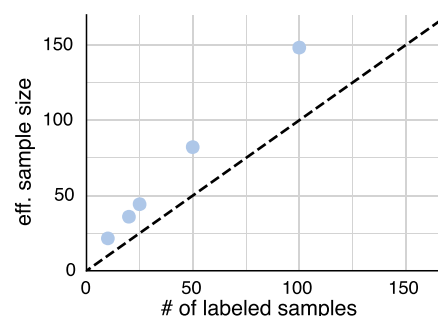


Figure 5.1: Eff. sample sizes of our approach vs. a classical test to infer the average win rate of `gpt-3.5-turbo` against other LLMs in the Chatbot Arena [43].

out compromising statistical validity. Intuitively, we use the limited human data in order to measure the bias of the synthetic data. Then, we evaluate the model on the synthetic data and correct the bias using this estimate. The core statistical tool used for this debiasing is called prediction-powered inference (PPI) [7]; we will describe this tool in detail in the coming text. Figure 5.1 illustrates that our approach, applied to rank language models on the Chatbot Arena dataset [49, 176], effectively increases effective sample sizes by up to 50% compared to a classical test. This approach can improve both metric-based evaluations (Section 5.2) and pairwise-comparison-based evaluations (Section 5.3), and can readily be applied using an existing Python software.¹ We will include code snippets throughout for producing more precise unbiased evaluations. These lower-variance evaluations are also accompanied by confidence intervals.

5.1.1 Related Work

Autoevaluation has been a subject of interest, particularly in language modeling, well before the current wave of progress in machine learning [1, 50, 71]. Since the development of powerful machine learning systems such as `gpt-4`, the accuracy of the annotations that these systems produce has started to approach that of humans [176], giving substantial credence to autoevaluation as an alternative to human evaluations. The prohibitive cost of human annotation has also encouraged the development of automatic metrics used to evaluate model performance without human aid [102, 121], representing a distinct but related approach to autoevaluation. Automatic metrics can be computed on the fly, rely on more data points and are hence less noisy, which can be more informative than human evaluations when the latter are scarce [162]. Standard autoevaluation methods are generally ad hoc, and resulting estimates of model performance can systematically differ from those obtained by human evaluation [71]. In parallel, classical solutions for generating confidence intervals, such as rank-sets [2], cannot take advantage of the AI-generated data. It has been unclear how AI-generated data can be *combined* with human data to improve the quality of evaluations. Towards this end, [39] produced lower-variance estimates of machine translation performance by combining human preferences with automated metrics via control variates.

Prediction-powered inference (PPI) is a set of estimators that incorporate predictions from machine learning models [7] to get lower-variance estimators that remain unbiased. In our case, we employ an optimized variant, PPI++ [10], in order to estimate metrics using synthetic data. From a statistical perspective, PPI is closely related to the fields of multiple imputation and semiparametric inference, perhaps most notably the augmented inverse propensity weighting (AIPW) estimator [130, 147] (see [10] for a careful review). Indeed, we are not the first to notice this application of PPI; the work of Saad-Falcon et al. [134] describes an autoevaluation method for evaluating and ranking language models from pairwise comparisons for the purpose of retrieval-augmented generation. A preprint by Chatzi et al. [43], posted concurrently with ours, also considers the problem of ranking models from

¹https://github.com/aangelopoulos/ppi_py

pairwise comparisons, and constructs approximate rankings with coverage guarantees. Our approach is complementary to these existing works. Our specific contribution is to develop an instantiation of PPI that is practical and yields tight confidence intervals, is easy to implement using existing software, and is compatible with existing evaluation systems such as Chatbot Arena [49, 108]. Moreover, we evaluate our PPI method on real data. Along the way, we develop an interesting extension of the PPI algorithms to the case where the annotation model outputs not just a single synthetic Y , but a distribution over Y .

5.2 Autoevaluating Accuracy and other Metrics

We begin by describing how to use prediction-powered inference for estimating metrics. The most commonly used metrics are accuracy and loss, so we focus on these; however, our tools will be general and allow autoevaluation of any metric.

5.2.1 Defining the Goal

Basic notation We observe inputs X in some space \mathcal{X} , such as the space of natural images, natural language, and so on. We seek to predict labels Y in some space \mathcal{Y} , such as the space of classes, next tokens, actions, etc. Towards this end, let f_1, \dots, f_M denote M pretrained models mapping inputs in \mathcal{X} to label estimates in some third space $\widehat{\mathcal{Y}}$. We often have $\widehat{\mathcal{Y}} = \mathcal{Y}$, in which case the model directly outputs predictions of the label. However, we leave open the possibility that $\widehat{\mathcal{Y}}$ is some other space—such as the space of softmax scores in the case of classification. The appropriate output space for $f(X)$ will be easy to infer from context.

Metrics We will evaluate the performance of the models by estimating the expectation of some metric function $\phi: \widehat{\mathcal{Y}} \times \mathcal{Y} \rightarrow \mathbb{R}$; in other words, the *metric* of model m will be

$$\mu_m = \mathbb{E}[\phi(f_m(X), Y)]$$

for some metric function ϕ and every $m = 1, \dots, M$. We are interested in estimating the M -length vector $\mu = (\mu_1, \dots, \mu_M)$. For example, in the case of the accuracy, we would want to measure

$$\text{accuracy}_m = \mathbb{E}[\phi_{\text{acc}}(f_m(X), Y)], \text{ where } \phi_{\text{acc}}(y, y') = \begin{cases} 1 & y = y' \\ 0 & \text{otherwise,} \end{cases}$$

for every $m \in 1, \dots, M$.

Accuracy is not the only quantity that can be framed within this setup. As another example, when the predictors are multilabel classifiers, one performance metric of interest could be the average precision of the model, that is, $\phi_{AP}(\hat{y}, y) := \frac{|\hat{y} \cap y|}{|\hat{y}|}$. In the case of regression, μ_m could correspond to the mean squared or absolute error of model m , in which case $\phi(\hat{y}, y) := (\hat{y} - y)^2$ or $\phi(\hat{y}, y) := |\hat{y} - y|$, respectively. Finally, one can imagine estimating

multiple losses at once; for example, for the purpose of assessing fairness, one may want to evaluate accuracy across many groups.

Data We assume access to two datasets: a small human-annotated dataset, $\{(X_i, Y_i)\}_{i=1}^n$, and a large amount of unlabeled data points, $\{X_i^u\}_{i=1}^N$, whose ground-truth labels $\{Y_i^u\}_{i=1}^N$ are unavailable. Importantly, both datasets are i.i.d.; extensions to some limited non-i.i.d. regimes are handled in [7], but we will not discuss them here. One should think of the regime where $N \gg n$: we have far more synthetic labels than real ones. For both datasets and every model, we also assume access to a *synthetic label distribution* that approximates $p(Y | X)$. We denote $\{\tilde{P}_{i,m}\}_{i=1}^n$ and $\{\tilde{P}_{i,m}^u\}_{i=1}^N$ as the set of synthetic label distributions conditioned on the labeled and unlabeled input data points, respectively. For each i and m , we will use the notation $d\tilde{P}_{i,m}(y)$ to represent the estimated PDF or PMF evaluated at label y .

For the sake of intuition, we make a few remarks regarding this data generating process. First, the synthetic data distributions can be seen as distributions over labels produced by one or several “annotator models”, that can either be related or different from the models to evaluate. In the latter case, the synthetic label distribution, for a given input, is the same for each model f_1, \dots, f_M . We do not need the subscript m in this scenario, and can simply denote the synthetic label distribution as \tilde{P}_i . However, the general case of $\tilde{P}_{i,m}$ allows for each model to have a different annotator model, and possibly allow models to self-annotate, that is, to themselves produce synthetic labels. Second, we note that the framework we described applies directly to the case where the annotator model produces single predictions of Y instead of distributions, by setting up $d\tilde{P}_{i,m}(y)$ to be a delta function at the prediction (the distribution is entirely concentrated on the prediction of Y).

5.2.2 The Algorithm

We combine the labeled and unlabeled data to estimate μ . In particular, we seek to benefit from the large sample size of the automatically annotated dataset to produce an estimator with low variance, while ensuring that this estimator remains unbiased. We will begin with the case of estimating accuracy, and then generalize our algorithm to arbitrary metrics.

Warm-Up: Model Accuracy

The classical approach to estimating model accuracy is to compute the fraction of correct labels:

$$\hat{\mu}_m^{\text{classical}} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(\hat{Y}_{i,m} = Y_i),$$

where $\hat{Y}_{i,m} = \arg \max_y f_m(X_i)_y$ and $f_m(X_i)$ is the softmax output of model m . Instead, we propose estimating the accuracy of a classifier differently: by using the classifier’s own confidence on the unlabeled data as a signal of its accuracy. Let $p_{i,m} = f_m(X_i)_{\hat{Y}_{i,m}}$ denote the top softmax score of model m on labeled example i , and $p_{i,m}^u, \hat{Y}_{i,m}^u$ be defined analogously.

Snippet 1 Python code to produce CIs and point estimates for model accuracy. The variable meanings are explained in the code comments.

```

from ppi_py import ppi_mean_pointestimate, ppi_mean_ci

# y_labeled <- (n,) ground-truth values of Y_i on labeled dataset
# yhat_labeled <- (n,M) predicted values of Y_i for each model on labeled dataset
# p_labeled <- (n,M) top softmax scores of each model on labeled dataset
# p_unlabeled <- (N,M) top softmax scores of each model on unlabeled dataset
# alpha <- (float) error rate of confidence interval

corrects = (yhat_labeled == y_labeled[:,None]).astype(float)

hat_mu = ppi_mean_pointestimate(corrects, p_labeled, p_unlabeled)
ci_mu = ppi_mean_ci(corrects, p_labeled, p_unlabeled, alpha=alpha)

```

We will use the estimator

$$\hat{\mu}_m := \underbrace{\lambda \frac{1}{N} \sum_{i=1}^N p_{i,m}^u}_{\text{estimated accuracy}} + \underbrace{\frac{1}{n} \sum_{i=1}^n (\mathbb{1} \{ \hat{Y}_{i,m} = Y_i \} - \lambda p_{i,m})}_{\text{bias correction}}, \quad (5.1)$$

where λ is a tuning parameter—for the time being, think of $\lambda = 1$. The above estimator decomposes into two natural components. Interpreting the top softmax score as the probability of correctness, the first term captures the model’s internal estimate of its accuracy on the unlabeled data. The second term is the bias of the first term.

This estimator has two beneficial properties: *unbiasedness* and *variance reduction*. Unbiasedness means that $\mathbb{E}[\hat{\mu}] = \mu$. This implies that the inclusion of machine learning predictions in our estimator does not introduce systematic errors for estimating the accuracy. Variance reduction means that the use of synthetic data reduces the variance of our estimator: $\text{Var}(\hat{\mu}_m) \leq \text{Var}(\hat{\mu}_m^{\text{classical}})$.

This is formally true for the optimally chosen parameter λ ; indeed, the optimal choice of λ ensures that our estimator is always better than $\hat{\mu}^{\text{classical}}$ (in an asymptotic sense). See [10] for details and a formal proof; the software in Snippet 1 automatically calculates this parameter.

General Metrics

The approach we have presented for evaluating classifier accuracy is an instance of a more general framework for evaluating properties of machine learning models. In particular, we can use our annotator model to output an approximate expectation of each label $\{Y_i\}_{i=1}^n$ and

$\{Y_i^u\}_{i=1}^N$ in the following way:

$$\hat{\mathbb{E}}_{i,m} = \int_{y \in \mathcal{Y}} \phi(f_m(X_i), y) d\tilde{P}_i(y) \quad \hat{\mathbb{E}}_{i,m}^u = \int_{y \in \mathcal{Y}} \phi(f_m(X_i^u), y) d\tilde{P}_i^u(y).$$

This expression looks complicated, but it has a simple interpretation: the annotator model, given X_i , thinks the distribution of Y_i is $d\tilde{P}_i$, and we are simply calculating the expected metric under that estimated distribution. This explains the hats on the expectation symbols; these are not real expectations, but rather, estimated expectations according to the annotator model. Indeed, in the case of classification, we can directly see that, as is intuitive, the expected accuracy of the m th model on the i th data point is just equal to its top softmax score:

$$\hat{\mathbb{E}}_{i,m} = \sum_{y \in \mathcal{Y}} \phi_{acc}(\hat{Y}_{i,m}, y) d\tilde{P}_{i,m}(y) = d\tilde{P}_{i,m}(\hat{Y}_{i,m}) = p_{i,m}.$$

Along the same lines, our previous estimator can be generalized to the case of arbitrary metrics as

$$\hat{\mu}_m := \underbrace{\lambda \frac{1}{N} \sum_{i=1}^N \hat{\mathbb{E}}_{i,m}^u}_{\text{metric on synthetic data}} + \underbrace{\frac{1}{n} \sum_{i=1}^n (\phi(f_m(X_i), Y_i) - \lambda \hat{\mathbb{E}}_{i,m})}_{\text{bias correction}}. \quad (5.2)$$

The first sum in the above expression is the average metric predicted by model m over all synthetic labels. If the annotator model is near-perfect and N is large, then this term will almost exactly recover the metric. However, if the synthetic label distribution is not good, this can bias our estimate of the metric. The second term corrects this bias by calculating it on the labeled dataset and subtracting it off.

Returning to the role of the tuning parameter: $\lambda \in [0, 1]$ is a discount factor on our synthetic data. When the synthetic data is very good, we can set $\lambda = 1$; when it is bad, setting $\lambda = 0$ will throw it away entirely. One can asymptotically optimize the variance of $\hat{\mu}$ in order to set λ , as in [10].

Again, it is straightforward to see that for any fixed value of λ , our estimator in (5.2) is unbiased, meaning $\mathbb{E}[\hat{\mu}] = \mu$, and will be strictly lower-variance than its classical counterpart when λ is optimally chosen.

Variance and Confidence Intervals

As we have explained above, the main benefit of AutoEval is to reduce the number of human-labeled data points to achieve a particular variance. We can formalize this by analyzing the variance of $\hat{\mu}_m$ and $\hat{\mu}_m^{\text{classical}}$. In particular, we can write the covariance matrix of $\hat{\mu}$ as

$$\frac{1}{n} V = \frac{1}{N} \lambda^2 \text{Cov}(L_i^u) + \frac{1}{n} \text{Cov}(\Delta_i^\lambda),$$

where $\Delta_i^\lambda = \left(\phi(f_1(X_i), Y_i) - \lambda \hat{\mathbb{E}}_{i,1}, \dots, \phi(f_M(X_i), Y_i) - \lambda \hat{\mathbb{E}}_{i,M} \right)$.

This expression admits a simple plug-in estimator; it also indicates that we should pick λ to minimize V in the appropriate sense. It also allows for the production of non-asymptotic confidence intervals using concentration. We opt to use asymptotic confidence intervals for μ . In particular, we have that as n and N approach infinity,

$$\sqrt{n} \widehat{V}^{-1/2} (\hat{\mu} - \mu) \rightarrow \mathcal{N}(0, \mathbb{I}_M),$$

where $\widehat{V} = \frac{n}{N} \lambda^2 \widehat{\text{Cov}}(L_i^u) + \widehat{\text{Cov}}(\Delta_i)$, $L_i^u = (\hat{\mathbb{E}}_{i,1}^u, \dots, \hat{\mathbb{E}}_{i,M}^u)$. \widehat{V} is the plug-in estimator of V , computable from the data.

Note that when $\lambda = 0$, we exactly recover $\hat{\mu}^{\text{classical}}$ —but this may not be the parameter that minimizes the variance \widehat{V} . Indeed, we can explicitly choose λ to minimize the variance (an explicit expression for this estimate can be found in [10]).

Another beneficial aspect of the asymptotic analysis is that it allows us to construct confidence intervals with which we can reliably rank models. For example, coordinatewise, the following is an asymptotically valid $1 - \alpha$ confidence interval *marginally* for each $\hat{\mu}_m$:

$$\mathcal{C}_m = \left(\hat{\mu}_m \pm \frac{z_{1-\alpha/2} \widehat{V}_{m,m}}{\sqrt{n}} \right). \quad (5.3)$$

The above interval comes with the following (standard) guarantee for all $m = 1, \dots, M$: $\lim_{n, N \rightarrow \infty} \mathbb{P}(\mu_m \in \mathcal{C}_m) = 1 - \alpha$.

As an alternative to producing confidence intervals for a single coordinate μ_m based on Equation (5.3), we might want to create confidence sets that contains the entire vector μ , that is, simultaneously valid intervals. The simultaneous interval can be constructed using the chi-squared distribution as

$$\mathcal{C}^X = \left\{ \mu : n \left\| \widehat{V}^{-1/2} (\hat{\mu} - \mu) \right\|_2^2 \leq \chi_{1-\alpha, M}^2 \right\},$$

where $\chi_{1-\alpha, M}^2$ denotes the $1 - \alpha$ quantile of the chi-squared distribution with M degrees of freedom. This interval has the following (standard) guarantee:

$$\lim_{n, N \rightarrow \infty} \mathbb{P}(\mu \in \mathcal{C}^X) = 1 - \alpha,$$

and thus, it can be used to rank the models by checking whether the m and m' coordinates of \mathcal{C}^X overlap for each model m and m' in $1, \dots, M$.

5.2.3 Application to Rank computer Vision Models

We applied the described methodology applies for evaluating computer vision models. We considered five trained computer vision models (ResNet-18, ResNet-34, ResNet-50, ResNet-101,

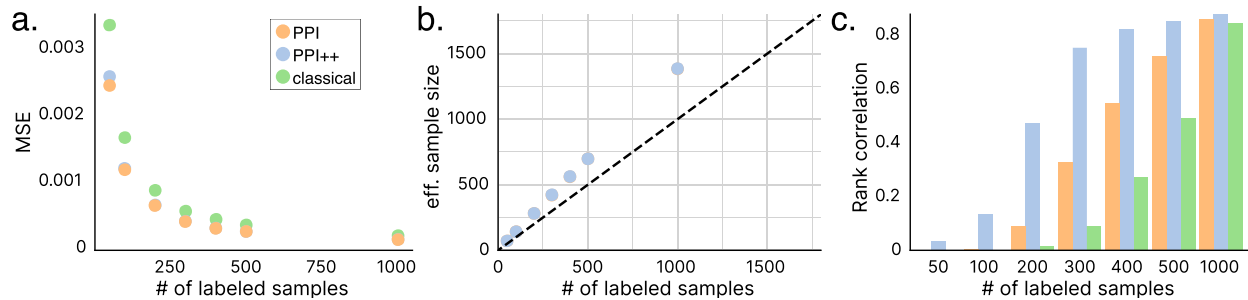


Figure 5.2: **ImageNet experiment.** For every approach, we built confidence intervals around the average accuracy of different ResNet architectures. **a.** MSE of the point estimates of the model accuracies. **b.** ESS of *PPI* and *PPI++* against the classical approach. **c.** Correlation between the estimated and true model rankings. Here, and in all following figures, obtained metrics are averaged across 250 random splits of the validation data into labeled and unlabeled data.

and ResNet-152) optimized over the training set of ImageNet and sourced from PyTorch [123]. We considered the task of estimating their accuracy on the validation set of ImageNet in a low-data regime.

We considered two different approaches to estimate the accuracy of these models. The first is referred to as PPI [7], and corresponds to (5.1) with $\lambda = 1$. The second strategy, PPI++ [10] optimizes λ to minimize the variance. These approaches were benchmarked against $\hat{\mu}^{\text{classical}}$ along with a standard z-test confidence interval.

To reflect a low-data regime, we randomly sampled a small number n of observations to be used as labeled data points available for these approaches. The rest of the observations in the validation data were used as unlabeled data points for PPI and PPI++. Our synthetic label distribution $d\tilde{P}_{i,m}$ is the softmax vector of model m on data point i , and $d\tilde{P}_{i,m}^u$ is analogous. Ground-truth model accuracies were computed as the mean accuracies evaluated over the entire validation dataset.

The mean-squared error of our estimates of the model accuracies improved over the classical baseline (Figure 5.2a). Both PPI and PPI++ had lower mean-squared errors than the baseline, no matter the size of the labeled set. Little to no difference was observed between PPI and PPI++, which probably means that the imputed accuracy scores are reliable proxies for the true quantities. Our approach hence provided more accurate point estimates of the model accuracies. When uncertainty quantification does matter, PPI and PPI++ provided calibrated confidence intervals across all labeled set sizes, and produced tighter confidence intervals than the baseline (Supplement 5.4.1).

The benefit of using unlabeled data can be measured by computing the effective sample size (ESS) of PPI and PPI++ relative to the classical approach (Figure 5.2b). This value can

be interpreted as the equivalent number of labeled data points for the classical approach that would be required to achieve the same level of precision as PPI or PPI++. Our ESS exceeds that of the classical approach by approximately 50%, which demonstrates the utility of unlabeled data for evaluating model performance.

Here, and in the other experiments, we also evaluated our approach for the purpose of model ranking, by ranking models based on their confidence intervals after Bonferroni correction. Models with overlapping confidence intervals were considered tied. Figure 5.2c shows the correlation of the estimated model ranks with the ground truth ranking (computed on all data) for different n and averaged across labeled-unlabeled data splits. This experiment showed dramatic differences between the approaches. PPI++ showed much stronger correlations with the ground truth than the other approaches, meaning that its rankings were more accurate and less prone to ties.

5.2.4 Application to Evaluate Protein Fitness Prediction Models

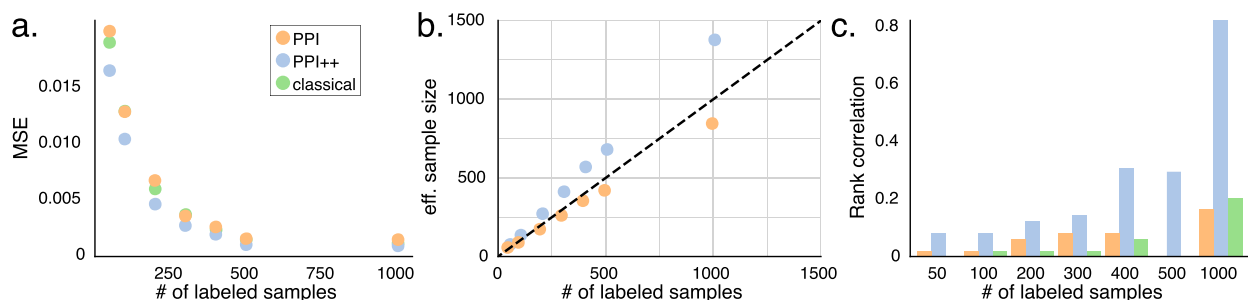


Figure 5.3: **Protein fitness experiment** for building confidence intervals and point estimates for the Pearson correlation of seven protein language models with the experimental fitness scores, using a held-out model to produce synthetic labels. **a.** MSE of the point estimates of the model correlations. **b.** ESS of *PPI* and *PPI++* against the classical approach. **c.** Correlation between the estimated and true model rankings.

We also used AutoEval to rank regression models, and more specifically, protein fitness prediction models. Protein fitness prediction is a crucial task in computational biology, aiming to predict the biological relevance of protein mutations based on their amino acid sequences. The recent development of deep learning models for protein language modeling has enabled the emergence of powerful models, trained on millions of protein sequences, used to predict protein fitness in a zero-shot manner [114]. Unfortunately, evaluating these models for a specific task remains challenging due to the scarcity of experimental data that can be used for evaluation, typically requiring expensive, time-consuming, and poorly scalable wet-lab experiments [95, 128].

We applied AutoEval on ProteinGym [116], which gathers several assays containing both experimental fitness measurements (that can be used as ground-truth labels), and predicted

fitness scores from a variety of models. We focused on ranking protein language models for predicting the fitness of mutations in the IgG-binding domain mutations of protein G based on an assay of $N = 536,962$ pairwise mutations [117]. We considered a scenario where one aims to select the best model for zero-shot fitness prediction for a specific protein, using a small experimental dataset and a large set of potential mutations for which fitness is not measured. To rank models, we focused on evaluating the Pearson correlation $r_m = \mathbb{E}[Y f_m(X)]$, where Y, X, f_m are the experimental fitness, the protein sequence, and the m -th fitness predictor, respectively, assuming that Y and $f_m(X)$ have zero mean and unit variance.² Unlike the ImageNet experiment, however, models did not produce internal confidence scores like the softmax scores of the classification models. Instead, we relied on a held-out model used as annotator, VESPA [112], to produce synthetic labels and allow us to leverage the unlabeled data. In this context, the general estimator in (5.2) takes the simple form

$$\hat{\mu}_m = \lambda \frac{1}{N} \sum_{i=1}^N f_m(X_i^u) f_{\text{VESPA}}(X_i^u) + \frac{1}{n} \sum_{i=1}^n (f_m(X_i) Y_i - \lambda f_m(X_i) f_{\text{VESPA}}(X_i))$$

The results of this experiment are shown in Figure 5.3. The effective sample sizes of PPI++ were systematically higher than the classical approach (Figure 5.3b), by approximately 50%. Furthermore, the ranks obtained by our approach were also much closer to the true model ranks than the classical approach (Figure 5.3c), with a five-fold improvement for $n = 1000$.

PPI++ confidence intervals for models’ correlations with the experimental fitness scores were also slightly tighter than the classical approach, yet remained calibrated (Supplement 5.4.1). The PPI estimator performed worse than the classical approach. This is a known issue of this estimator, that PPI++ mitigates.

A question remains: how good does the annotator model need to be to allow AutoEval to work well? Figure 5.4 compares the effective sample size of PPI++ obtained with different annotator models. As expected, the better the annotator model, the higher the effective sample size. We importantly note that even with a very poor annotator model, PPI++ performs at least as well as the classical approach. When the annotator labels do not correlate with the true labels, PPI++ falls back to the classical approach ($\lambda = 0$), effectively ignoring the synthetic labels. That being said, we observe that even mediocre annotator models, such as CARP, provide a 10% increase in effective sample size compared to the classical approach. Altogether, these observations suggest that AutoEval can provide

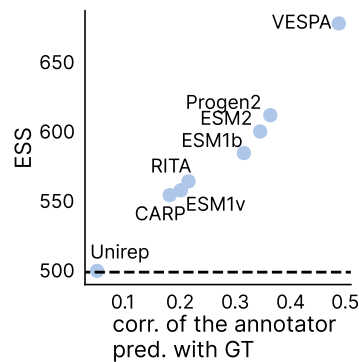


Figure 5.4: ESS of PPI++ against annotator model performance for $n = 500$ labeled points. The horizontal line denotes the ESS of classical.

²In practice, we standardized ground-truth labels and model predictions.

better point estimates and tighter confidence intervals compared to the classical approach even when the annotator model is mediocre.

5.3 Evaluating Model Performance from Pairwise Comparisons

Characterizing the absolute performance of ML models for the purpose of ranking them is challenging. The previous section described a methodology to compare models based on a common performance metric. Unfortunately, metrics serving as proxies for model performance might either not exist, or diverge from human judgment [88].

In such cases, assessing *relative* model performance might be more appropriate. This can typically be done by comparing different model predictions to each other. The Chatbot Arena project [108], for instance, allows human annotators to state preferences over different LLM predictions to the same prompt. Comparison-based evaluation is also an exciting opportunity for autoevaluation [176]. In particular, an external LLM, prompted to serve as an annotator, agrees with human annotators with high fidelity. Still, it is unclear how biased an AI annotator might be, which drastically limits the usefulness of the validation data it produces. This section describes how to leverage such AI-generated preferences while making statistically valid inferences about model performance.

5.3.1 A Model to Assess Relative Performance

The canonical model for assessing relative performance of models based on pairwise comparisons, as in a tournament, is called the Bradley-Terry (BT) model [27, 69, 173]. The BT model is used in the Chatbot Arena [49], by the World Chess Federation, the European Go Federation, and many other competitive organizations as a tool for ranking players.

Now we describe the BT model. Imagine, among M models, we are trying to compare the strength of model A to the strength of model B . Towards this end, we give a prompt Q to both models, and they give us an answer. We show this answer to a human, who gives us $Y = 1$ if the answer of model B is better than the answer of model A , and vice versa. The assumption of the BT model is that Y follows a logistic relationship,

$$P(Y = 1 \mid A, B) = \frac{1}{1 + e^{\zeta_A - \zeta_B}}, \quad (5.4)$$

with some parameter vector ζ of length M , whose entries are called the *Bradley-Terry coefficients*. Each model m has a BT coefficient ζ_m which, when large relative to the other coefficients, signifies that it is more likely to win the pairwise comparison. (Also, because the model in (5.4) is invariant to addition of a constant to every coordinate of ζ , we can, without loss of generality, set $\zeta_1 = 0$, making the model identifiable.)

It is well-known that, given a labeled dataset of n pairwise comparisons, $\{A_i, B_i, Q_i, Y_i\}$, the maximum-likelihood estimator of the BT coefficients is a logistic regression [84]. Let X_i be the vector of all zeros except at indexes A_i and B_i , where it is -1 and 1 respectively. The logistic regression estimate of the BT coefficients is

$$\hat{\zeta}^{\text{classical}} = \arg \min_{\zeta \in \mathbb{R}^{M-1}, \zeta_1=0} \frac{1}{n} \sum_{i=1}^n \ell_{\zeta}(X_i, Y_i),$$

where ℓ is the binary cross-entropy loss.

5.3.2 Autoevaluation of Relative Performance

Prediction-powered inference can be applied out-of-the-box to the BT model, making it possible to leverage large numbers of AI-generated preferences while controlling for their potential bias. In addition to the set of human preferences defined above, additionally define the unlabeled dataset $\{(A_i^u, B_i^u, Q_i^u)\}_{i=1}^N$. On both the labeled and unlabeled datasets, we have the prompt and both models' answers; we use `gpt-4` in place of the human to choose between the answers. This gives us a prediction \hat{Y}_i and \hat{Y}_i^u of the pairwise comparison on both datasets. The PPI++ estimator of the BT coefficients is given by

$$\hat{\zeta} = \arg \min_{\substack{\zeta \in \mathbb{R}^{M-1} \\ \zeta_1=0}} \frac{1}{n} \sum_{i=1}^n (\ell_{\zeta}(X_i, Y_i) - \lambda \ell_{\zeta}(X_i, \hat{Y}_i)) + \frac{\lambda}{N} \sum_{i=1}^N \ell_{\zeta}(X_i^u, \hat{Y}_i^u),$$

where $\lambda \in [0, 1]$ controls the weight we give to the AI-generated preferences. Although this estimator departs from the arguments given in Section 5.2, it has a very similar interpretation; it constructs an unbiased and lower-variance loss function for the true logistic regression, and then minimizes it.

The resulting BT coefficient estimates have the same appealing properties as above. In particular, they are unbiased for any fixed λ , and one can construct confidence intervals around them using PPI and PPI++; see [7, 10] for this and other generalized linear models, as well as methods for optimally choosing λ . Snippet 2 provides Python code to produce these CIs.

5.3.3 Autoevaluation of LLMs from Pairwise Preferences

We evaluated our approach on the Chatbot Arena dataset [49, 176]. Each observation contains an initial prompt, responses from two different LLMs, and a binary preference over these responses, either from a human expert or from another LLM, `gpt-4`. Conveniently, the same prompts are used to collect both types of preferences, allowing us to easily benchmark our approach. In total, the dataset contains paired expert and `gpt-4` preferences for 3,355 data points, each corresponding to a question, as well as answers from two different LLMs (out of six in total).

Snippet 2 Python code to produce CIs for the Bradley-Terry coefficients (without multiplicity correction). The variable meanings are explained in the code comments. For clarity, the matrix `X_labeled` has one row per pairwise comparison. The i th row is a two-hot vector, with -1 at position A_i and $+1$ at position B_i . The matrix `X_unlabeled` is analogous. Note that `X_labeled` and `X_unlabeled` have only $M - 1$ columns, since ζ_1 does not need to be estimated.

```

from ppi_py import ppi_logistic_pointestimate, ppi_logistic_ci

# X_labeled <- (n,M-1) one row per battle; -1 for model A, 1 for model B, 0 else
# y_labeled <- (n,) 0 if model A wins, 1 if model B wins
# yhat_labeled <- (n,) predicted values of Y_i on labeled dataset
# X_unlabeled <- (N,M-1) one row per battle; -1 for model A, 1 for model B, 0 else
# yhat_unlabeled <- (n,) predicted values of Y_i on unlabeled dataset
# alpha <- (float) error rate of confidence interval

hat_zeta = ppi_logistic_pointestimate(
    X_labeled, y_labeled, yhat_labeled,
    X_unlabeled, yhat_unlabeled
)

ci_zeta = ppi_logistic_ci(
    X_labeled, y_labeled, yhat_labeled,
    X_unlabeled, yhat_unlabeled, alpha=alpha
)

```

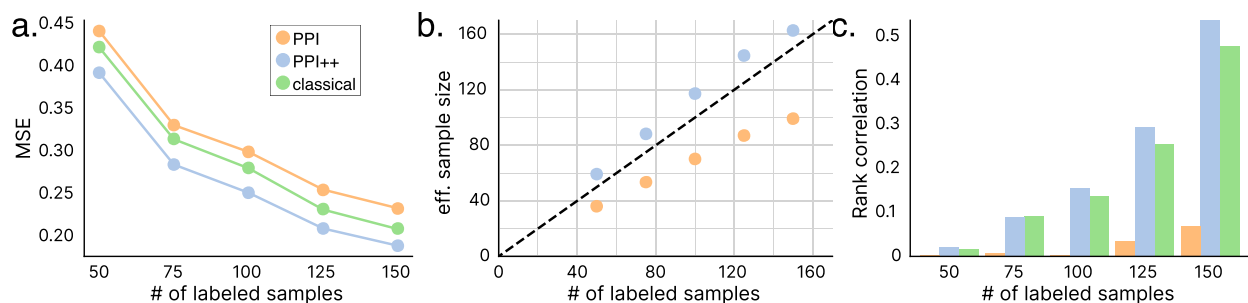


Figure 5.5: **LLM experiment** for building confidence intervals and point estimates for the BT coefficients of different LLMs. **a.** MSE of the point estimates of the BT coefficients. **b.** ESS of PPI and PPI++ against the classical approach. **c.** Correlation between the estimated and true model rankings.

We observed that the BT coefficients were better estimated by PPI++ than by the classical approach, hinting that the point estimates of AutoEval are more accurate (Figure 5.5a). We also observed ESS showing a 20 to 25% improvement over the classical approach (Figure 5.5b). Finally, we observed that the estimated rankings of the models were more correlated with the true rankings when using PPI++ (Figure 5.5c). Supplement 5.4.1 and Figure 5.1 provide results for the simpler task of estimating the win rate of `gpt-3.5-turbo` against other LLMs, showing again a substantial improvement over the classical approach due to the use of `gpt-4` preferences.

5.4 Supplementary Material

Experimental details

Data acquisition and preprocessing

ImageNet We downloaded model weights from Pytorch’s model zoo for the different ResNet models, trained on the training set of ImageNet. We then computed the different models’ predictions on the validation set of ImageNet on a high-performance computing cluster.

Protein fitness We relied on ProteinGym³ to access both the ground-truth fitness values and the predictions of the different protein language models for a specific assay corresponding to IgG-binding domain mutations of protein G [117] (`SPG1_STRSG_01son_2014`). All fitness scores were normalized as a preprocessing step.

LLM We considered the MT-bench dataset [176], containing more than 3K human preferences over pairs of LLM answers to the same prompts, along with `gpt-4` preferences for the same exchanges, for a total of six LLMs (`gpt-4`, `gpt-3.5-turbo`, `claude-v1`, `vicuna-13b-v1.2`, `alpaca-13b`, and `llama-13b`) The data contains questions/exchanges that were evaluated by multiple human annotators; we used the average of these annotations as the ground truth.

Methodological details

Monte Carlo trials In all experiments, we randomly split the data into labeled and unlabeled sets 250 times, and computed all point estimates in the main text and in this supplementary material as the average estimate over these splits.

Model ranking To rank models with the different estimators, we computed 90% confidence intervals for the different approaches after Bonferroni correction. Models with overlapping confidence intervals were assigned the same rank.

³<https://github.com/OATML-Markslab/ProteinGym>

Experimental setup

All AutoEval experiments were run on a workstation with 12th generation Intel (R)Core (TM) i9-12900KF, 128GB of RAM, and on a compute cluster relying on CPU nodes with four cores. We relied on the Python package `ppi_py` [7], except for the LLM experiment, for which we relied on Jax [26] to implement PPI and PPI++ for the Bradley-Terry model.

Table 5.1: Running times for the different experiments (in seconds). Displayed are the times to produce confidence intervals for PPI++, PPI, and the classical approach on a single labeled-unlabeled split.

	ImageNet	ProteinGym	LLM
Execution time	$2.3 \times 10^{-2} s$	$2.6 \times 10^{-1} s$	8.4s

5.4.1 Additional experiments

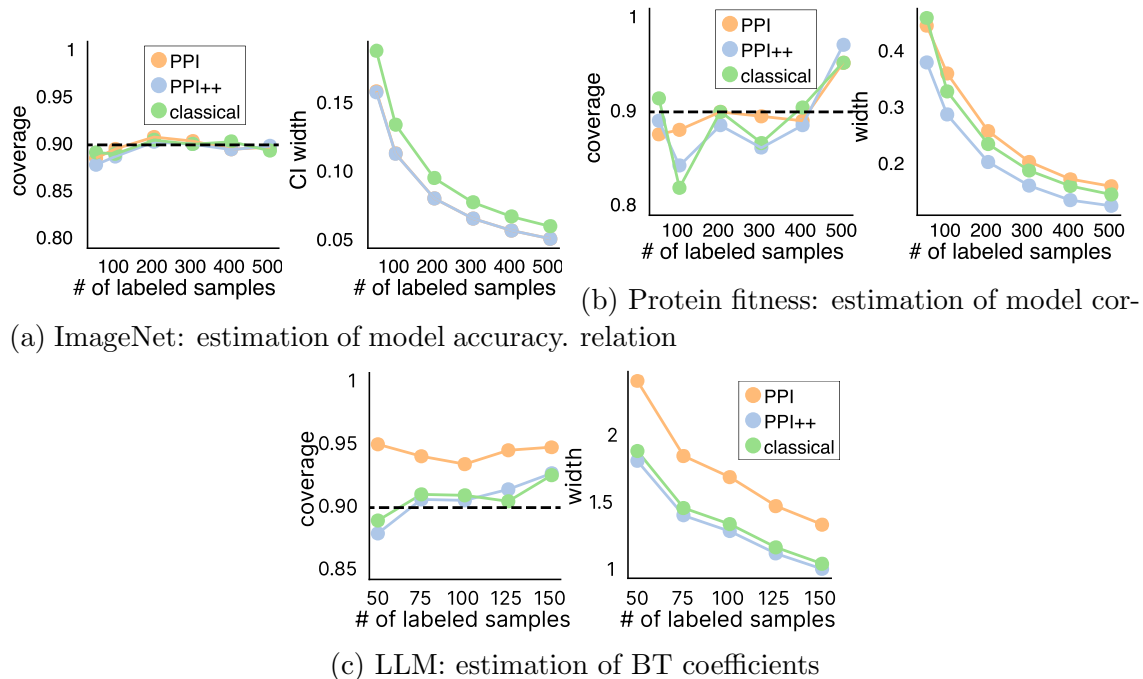


Figure S1: **Interval metrics for the different experiments.** **a.** and **b.** Coverage and width of the 90%-confidence intervals (without multiplicity correction). Each experiment is described in the main text and focuses on the estimation of a different metric.

This section provides additional experiments for the different experiments presented in the main text.

Interval metrics

We here show that the proposed methodology provides *calibrated* and *tight* confidence intervals for the various measurements of model performance described in the main text. Figure S1 displays the coverage and width of the 90%-confidence intervals for the different experiments, all of which show that both PPI and PPI++ are calibrated, as well as showing that PPI++ provides tighter confidence intervals than the classical approach.

Average win rate in the LLM experiment

As an alternative to the BT model, our approach can also be used to estimate a simpler metric, the average win rate of a model against other models, which corresponds to the probability that the model wins a pairwise comparison. Figures 5.1 and S2 respectively show the ESS and MSE for the estimation of the average win rate of `gpt-3.5-turbo` against other LLMs in the Chatbot Arena, both showing that AutoEval compares favorably to the classical approach.

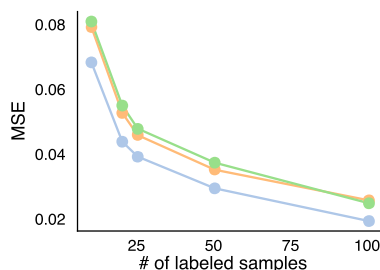


Figure S2: Mean-squared errors for the estimation of the average win rate of `gpt-3.5-turbo` against other LLMs in the Chatbot Arena.

Bibliography

- [1] Mayank Agarwal, Ritika Kalia, Vedant Bahel, and Achamma Thomas. Autoeval: An NLP approach for automatic test evaluation system. In *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 1–6. IEEE, 2021.
- [2] Diaa Al Mohamad, Erik van Zwet, Aldo Solari, and Jelle Goeman. Simultaneous confidence intervals for ranks using the partitioning principle. *Electronic Journal of Statistics*, 15:2608–2646, 2021.
- [3] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.
- [4] Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.
- [5] Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.
- [6] 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*, 2021.
- [7] Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnic. Prediction-powered inference. *Science*, 382:669–674, 2023.
- [8] Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnic. `ppi-py`: A python package for scientific discovery using machine learning. *GitHub*, 2023.
- [9] Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

- [10] Anastasios N Angelopoulos, John C Duchi, and Tijana Zrnic. Ppi++: Efficient prediction-powered inference. *arXiv preprint arXiv:2311.01453*, 2023.
- [11] Anastasios N Angelopoulos, Amit P Kohli, Stephen Bates, Michael I Jordan, Jitendra Malik, Thayer Alshaabi, Srigokul Upadhyayula, and Yaniv Romano. Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. *arXiv preprint arXiv:2202.05265*, 2022.
- [12] 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*, 2022.
- [13] Anastasios Nikolas Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2021.
- [14] 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*, 2021.
- [15] Martin Anthony and John Shawe-Taylor. A result of vapnik with applications. *Discrete Applied Mathematics*, 47(3):207–217, 1993.
- [16] David Azriel, Lawrence D. Brown, Michael Sklar, Richard Berk, Andreas Buja, and Linda Zhao. Semi-supervised linear regression. *Journal of the American Statistical Association*, 117(540):2238–2251, 2022.
- [17] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253, 2017.
- [18] Rina Barber, Emmanuel Candès, Aaditya Ramdas, and Ryan Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference*, 10, 08 2020.
- [19] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *arXiv preprint arXiv:2202.13415*, 2022.
- [20] Inigo Barrio-Hernandez, Jingsi Yeo, Jürgen Jänes, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *bioRxiv*, pages 2023–03, 2023.
- [21] Hamsa Bastani. Predicting with proxies: Transfer learning in high dimension. *Management Science*, 67(5):2964–2984, 2021.
- [22] Osbert Bastani, Varun Gupta, Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Practical adversarial multivald conformal prediction. *Advances in Neural Information Processing Systems*, 35:29362–29373, 2022.

- [23] Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6), September 2021.
- [24] Aadyot Bhatnagar, Huan Wang, Caiming Xiong, and Yu Bai. Improved online conformal prediction via strongly adaptive online learning. *arXiv preprint arXiv:2302.07869*, 2023.
- [25] Isabell Bludau, Sander Willems, Wen-Feng Zeng, Maximilian T Strauss, Fynn M Hansen, Maria C Tanzer, Ozge Karayel, Brenda A Schulman, and Matthias Mann. The structural context of posttranslational modifications at a proteome-wide scale. *PLoS Biology*, 20(5):e3001636, 2022.
- [26] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. 2018.
- [27] Ralph Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- [28] F Jay Breidt and Jean D Opsomer. Model-assisted survey estimation with modern prediction techniques. *Statistical Science*, 32(2):190–205, 2017.
- [29] Sébastien Bubeck. Introduction to online optimization. *Lecture notes*, 2:1–86, 2011.
- [30] Andreas Buja, Lawrence Brown, Richard Berk, Edward George, Emil Pitkin, Mikhail Traskin, Kai Zhang, and Linda Zhao. Models as approximations I: Consequences illustrated with linear regression. *Statistical Science*, 34(4):523–544, 2019.
- [31] Monimoy Bujarbaruah, Xiaojing Zhang, Marko Tanaskovic, and Francesco Borrelli. Adaptive stochastic mpc under time-varying uncertainty. *IEEE Transactions on Automatic Control*, 66(6):2840–2845, 2020.
- [32] Eric L Bullock, Curtis E Woodcock, Carlos Souza Jr, and Pontus Olofsson. Satellite-based estimates reveal widespread forest degradation in the Amazon. *Global Change Biology*, 26(5):2956–2969, 2020.
- [33] Feiyang Cai and Xenofon Koutsoukos. Real-time out-of-distribution detection in learning-enabled cyber-physical systems. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, pages 174–183. IEEE, 2020.
- [34] Marco C Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.

- [35] 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.
- [36] Clément L Canonne. A short note on learning discrete distributions. *arXiv preprint arXiv:2002.11457*, 2020.
- [37] Raymond J Carroll, David Ruppert, Leonard A Stefanski, and Ciprian M Crainiceanu. *Measurement Error in Nonlinear Models: A Modern Perspective*. Chapman and Hall/CRC, 2006.
- [38] Claes M Cassel, Carl E Särndal, and Jan H Wretman. Some results on generalized difference estimation and generalized regression estimation for finite populations. *Biometrika*, 63(3):615–620, 1976.
- [39] Arun Tejasvi Chaganty, Stephen Mussman, and Percy Liang. The price of debiasing automatic metrics in natural language evaluation. *arXiv preprint arXiv:1807.02202*, 2018.
- [40] Abhishek Chakraborty and Tianxi Cai. Efficient and adaptive linear regression in semi-supervised settings. *Annals of Statistics*, 46(4):1541–1572, 2018.
- [41] Abhishek Chakraborty, Guorong Dai, and Raymond J Carroll. Semi-supervised quantile estimation: Robust and efficient inference in high dimensional settings. *arXiv preprint arXiv:2201.10208*, 2022.
- [42] Abhishek Chakraborty, Guorong Dai, and Eric Tchetgen Tchetgen. A general framework for treatment effect estimation in semi-supervised and high dimensional settings. *arXiv preprint arXiv:2201.00468*, 2022.
- [43] Ivi Chatzi, Eleni Straitouri, Suhas Thejaswi, and Manuel Gomez Rodriguez. Prediction-powered ranking of large language models. *arXiv preprint arXiv:2402.17826*, 2024.
- [44] Jinbo Chen and Norman E Breslow. Semiparametric efficient estimation for the auxiliary outcome problem with the conditional mean model. *Canadian Journal of Statistics*, 32(4):359–372, 2004.
- [45] Song Xi Chen, Denis H Y Leung, and Jing Qin. Information recovery in a study with surrogate endpoints. *Journal of the American Statistical Association*, 98(464):1052–1062, 2003.
- [46] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

- [47] Xiaohong Chen, Han Hong, and Elie Tamer. Measurement error models with auxiliary data. *The Review of Economic Studies*, 72(2):343–366, 2005.
- [48] Yuxiao Chen, Ugo Rosolia, Chuchu Fan, Aaron Ames, and Richard Murray. Reactive motion planning with probabilistic safety guarantees. In *Conference on Robot Learning*, pages 1958–1970. PMLR, 2021.
- [49] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. *arXiv preprint arXiv:2403.04132*, 2024.
- [50] Simon Corston-Oliver, Michael Gamon, and Chris Brockett. A machine learning approach to the automatic evaluation of machine translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 148–155, 2001.
- [51] 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.
- [52] A Philip Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.
- [53] Oscar de Groot, Laura Ferranti, Dariu Gavrila, and Javier Alonso-Mora. Scenario-based motion planning with bounded probability of collision. *arXiv preprint arXiv:2307.01070*, 2023.
- [54] John DeHardt. Generalizations of the Glivenko-Cantelli theorem. *The Annals of Mathematical Statistics*, 42(6):2050–2055, 1971.
- [55] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.
- [56] Siyi Deng, Yang Ning, Jiwei Zhao, and Heping Zhang. Optimal and safe estimation for high-dimensional semi-supervised learning. *arXiv preprint arXiv:2011.14185*, 2020.
- [57] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34:6478–6490, 2021.
- [58] Anushri Dixit, Lars Lindemann, Skylar X Wei, Matthew Cleaveland, George J Pappas, and Joel W Burdick. Adaptive conformal prediction for motion planning among dynamic agents. In *Learning for Dynamics and Control Conference*, pages 300–314. PMLR, 2023.

- [59] Noel E Du Toit and Joel W Burdick. Robot motion planning in dynamic, uncertain environments. *IEEE Transactions on Robotics*, 28(1):101–115, 2011.
- [60] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.
- [61] Bradley Efron. *Exponential Families in Theory and Practice*. Cambridge University Press, 2022.
- [62] Paul Erdős. On the central limit theorem for samples from a finite population. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 4:49–61, 1959.
- [63] 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, 2020.
- [64] 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.
- [65] António Farinhas, Chrysoula Zerva, Dennis Ulmer, and André FT Martins. Non-exchangeable conformal risk control. *arXiv preprint arXiv:2310.01262*, 2023.
- [66] Shai Feldman, Liran Ringel, Stephen Bates, and Yaniv Romano. Achieving risk control in online learning settings. *Transactions on Machine Learning Research*, 2023.
- [67] Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. *arXiv preprint arXiv:2007.03114*, 2020.
- [68] Adam Fisch, Tal Schuster, Tommi Jaakkola, and Regina Barzilay. Conformal prediction sets with limited false positives. *arXiv preprint arXiv:2202.07650*, 2022.
- [69] Lester R Ford Jr. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8P2):28–33, 1957.
- [70] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.
- [71] Saurabh Garg, Sivaraman Balakrishnan, Zachary C Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234*, 2022.

- [72] Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.
- [73] Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems*, volume 34, pages 1660–1672. Curran Associates, Inc., 2021.
- [74] Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pages 45–87, 2020.
- [75] Leying Guan. Conformal prediction with localization. *arXiv:1908.08558*, 2020.
- [76] Thomas Hamelryck and Bernard Manderick. PDB file parser and structure class implemented in python. *Bioinformatics*, 19(17):2308–2310, November 2003.
- [77] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
- [78] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- [79] Elad Hazan and Karan Singh. Introduction to online nonstochastic control. *arXiv preprint arXiv:2211.09619*, 2022.
- [80] 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*, pages 770–778, 2016.
- [81] Thomas Höglund. Sampling from a finite population. A remainder term estimate. *Scandinavian Journal of Statistics*, pages 69–71, 1978.
- [82] Jue Hou, Zijian Guo, and Tianxi Cai. Surrogate assisted semi-supervised inference for high dimensional risk prediction. *arXiv preprint arXiv:2105.01264*, 2021.
- [83] Kai-Chieh Hsu, Haimin Hu, and Jaime Fernández Fisac. The safety filter: A unified view of safety-critical control in autonomous systems, 2023.
- [84] David R Hunter. MM algorithms for generalized Bradley-Terry models. *The annals of statistics*, 32(1):384–406, 2004.
- [85] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

- [86] Lilia M Iakoucheva, Predrag Radivojac, Celeste J Brown, Timothy R O'Connor, Jason G Sikes, Zoran Obradovic, and A Keith Dunker. The importance of intrinsic disorder for protein phosphorylation. *Nucleic acids research*, 32(3):1037–1049, 2004.
- [87] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.
- [88] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.
- [89] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.
- [90] Nathan Kallus and Xiaojie Mao. On the role of surrogates in the efficient estimation of treatment effects with limited outcome data. *arXiv preprint arXiv:2003.12408*, 2020.
- [91] 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)*, Online, November 2020. Association for Computational Linguistics.
- [92] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.
- [93] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 33–50, 1978.
- [94] 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.
- [95] Elodie Laine, Yasaman Karami, and Alessandra Carbone. GEMME: A simple and fast global epistatic model predicting mutational effects. *Molecular Biology and Evolution*, 36(11):2604–2619, August 2019.
- [96] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

- [97] Jouko Lampinen and Aki Vehtari. Bayesian approach for neural networks—review and case studies. *Neural networks*, 14(3):257–274, 2001.
- [98] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.
- [99] Erich Leo Lehmann. Some principles of the theory of testing hypotheses. In *Selected works of EL Lehmann*, pages 139–164. Springer, 2011.
- [100] 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.
- [101] Lihua Lei and Emmanuel J. Candès. Conformal inference of counterfactuals and individual treatment effects. *arXiv:2006.06138*, 2020.
- [102] Chin-Yew Lin and FJ Och. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir workshop*, 2004.
- [103] Haotian Lin and Matthew Reimherr. On transfer learning in functional linear regression. *arXiv preprint arXiv:2206.04277*, 2022.
- [104] Tsung-Yi Lin, Michael Maire, Serge 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*, pages 740–755. Springer, 2014.
- [105] Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 2023.
- [106] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning*, pages 3122–3130. PMLR, 2018.
- [107] Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons, 2019.
- [108] LMSYS. Chatbot Arena: Benchmarking LLMs in the wild. <https://chat.lmsys.org/>. Accessed: 2024-02-09.
- [109] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [110] Rachel Luo, Shengjia Zhao, Jonathan Kuck, Boris Ivanovic, Silvio Savarese, Edward Schmerling, and Marco Pavone. Sample-efficient safety assurances using conformal prediction. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 149–169. Springer, 2022.

- [111] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.
- [112] Céline Marquet, Michael Heinzinger, Tobias Olenyi, Christian Dallago, Kyra Erckert, Michael Bernhofer, Dmitrii Nechaev, and Burkhard Rost. Embeddings from protein language models predict conservation and variant effects. *Human Genetics*, 141(10):1629–1647, October 2022.
- [113] Pascal Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, pages 1269–1283, 1990.
- [114] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, page 2021.07.09.450648, November 2021.
- [115] Anish Muthali, Haotian Shen, Sampada Deglurkar, Michael H Lim, Rebecca Roelofs, Aleksandra Faust, and Claire Tomlin. Multi-agent reachability calibration with conformal prediction. *arXiv preprint arXiv:2304.00432*, 2023.
- [116] Pascal Notin, Aaron W Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Hansen Spinner, Nathan Rollins, Ada Shaw, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Rose Orenbuch, Yarin Gal, and Debora S Marks. ProteinGym: Large-Scale benchmarks for protein design and fitness prediction. *bioRxiv*, page 2023.12.07.570727, December 2023.
- [117] C Anders Olson, Nicholas C Wu, and Ren Sun. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current Biology*, 24(22):2643–2651, November 2014.
- [118] Robert J Olson, Alexi Shalapyonok, and Heidi M Sosik. An automated submersible flow cytometer for analyzing pico-and nanophytoplankton: FlowCytobot. *Deep Sea Research Part I: Oceanographic Research Papers*, 50(2):301–315, 2003.
- [119] Eric C Orenstein, Oscar Beijbom, Emily E Peacock, and Heidi M Sosik. WHOI-plankton-a large scale fine grained visual recognition benchmark dataset for plankton classification. *arXiv preprint arXiv:1510.00745*, 2015.
- [120] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pages 345–356. Springer, 2002.

- [121] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [122] 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.
- [123] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.
- [124] Margaret Sullivan Pepe. Inference using surrogate outcome data and a validation sample. *Biometrika*, 79(2):355–365, 1992.
- [125] 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.
- [126] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *Conference on Robot Learning*, 2023.
- [127] Tal Ridnik, Hussam Lawen, Asaf Noy, and Itamar Friedman. TResNet: High performance GPU-dedicated architecture. *arXiv:2003.13630*, 2020.
- [128] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, September 2018.
- [129] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14*, pages 549–565. Springer, 2016.
- [130] James M Robins and Andrea Rotnitzky. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*, 90(429):122–129, 1995.
- [131] James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994.

- [132] Yaniv Romano, Evan Patterson, and Emmanuel Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, volume 32, pages 3543–3553. NeurIPS, 2019.
- [133] 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., 2020.
- [134] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*, 2023.
- [135] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajec-tron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Com-puter Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.
- [136] 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.
- [137] Carl-Erik Särndal, Bengt Swensson, and Jan Wretman. *Model Assisted Survey Sam-pling*. Springer Science & Business Media, 1992.
- [138] Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *arXiv preprint arXiv:2207.07061*, 2022.
- [139] Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent acceler-ated inference via confident adaptive transformers. *arXiv preprint arXiv:2104.08803*, 2021.
- [140] Joseph O Sexton, Xiao-Peng Song, Min Feng, Praveen Noojipady, Anupam Anand, Chengquan Huang, Do-Hyung Kim, Kathrine M Collins, Saurabh Channan, Charlene DiMiceli, et al. Global, 30-m resolution continuous fields of tree cover: Landsat-based rescaling of modis vegetation continuous fields with lidar-based estimates of error. *International Journal of Digital Earth*, 6(5):427–448, 2013.
- [141] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar):371–421, 2008.
- [142] Rohan Sinha, Edward Schmerling, and Marco Pavone. Closing the loop on runtime monitors with fallback-safe mpc. *Conference on Decision and Control*, 2023.

- [143] 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.
- [144] Shanshan Song, Yuanyuan Lin, and Yong Zhou. A general m-estimation theory in semi-supervised framework. *Journal of the American Statistical Association*, pages 1–11, 2023.
- [145] Ye Tian and Yang Feng. Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, pages 1–14, 2022.
- [146] 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. NeurIPS, 2019.
- [147] Anastasios A Tsiatis. *Semiparametric Theory and Missing Data*. Springer, 2006.
- [148] Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Židek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, et al. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596, 2021.
- [149] UniProt Consortium. UniProt: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2015.
- [150] Eeshit Dhaval Vaishnav, Carl G de Boer, Jennifer Molinet, Moran Yassour, Lin Fan, Xian Adiconis, Dawn A Thompson, Joshua Z Levin, Francisco A Cubillos, and Aviv Regev. The evolution, evolvability and engineering of gene regulatory DNA. *Nature*, 603(7901):455–463, March 2022.
- [151] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018.
- [152] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior (60th Anniversary Commemorative Edition)*. Princeton university press, 2007.
- [153] Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian Conference on Machine Learning*, volume 25, pages 475–490, 2012.
- [154] Vladimir Vovk and Claus Bendtsen. Conformal predictive decision making. In *Conformal and Probabilistic Prediction and Applications*, pages 52–62. PMLR, 2018.
- [155] Vladimir Vovk, Alexander Gammerman, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *International Conference on Machine Learning*, pages 444–453, 1999.

- [156] Vladimir Vovk, Alexander Gammernan, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.
- [157] Kim P. Wabersich, Andrew J. Taylor, Jason J. Choi, Koushil Sreenath, Claire J. Tomlin, Aaron D. Ames, and Melanie N. Zeilinger. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.
- [158] Rahee Walambe, Nipun Agarwal, Swagatu Kale, and Vrunda Joshi. Optimal trajectory generation for car-type mobile robot using spline interpolation. *IFAC-PapersOnLine*, 49(1):601–606, 2016.
- [159] Siruo Wang, Tyler H McCormick, and Jeffrey T Leek. Methods for correcting inference based on outcomes predicted by machine learning. *Proceedings of the National Academy of Sciences*, 117(48):30266–30275, 2020.
- [160] Larry Wasserman and John Lafferty. Statistical analysis of semi-supervised regression. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.
- [161] Ian Waudby-Smith and Aaditya Ramdas. Variance-adaptive confidence sequences by betting. *arXiv:2010.09686*, 2020.
- [162] Johnny Tian-Zheng Wei and Robin Jia. The statistical advantage of automatic NLG metrics at the system level. *arXiv preprint arXiv:2105.12437*, 2021.
- [163] Halbert White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica: Journal of the Econometric Society*, pages 817–838, 1980.
- [164] Halbert White. Using least squares to approximate unknown regression functions. *International Economic Review*, pages 149–170, 1980.
- [165] Kyle W Willett, Chris J Lintott, Steven P Bamford, Karen L Masters, Brooke D Simmons, Kevin RV Casteels, Edward M Edmondson, Lucy F Fortson, Sugata Kaviraj, William C Keel, et al. Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 435(4):2835–2860, 2013.
- [166] Changbao Wu and Randy R Sitter. A model-calibration approach to using complete auxiliary information from survey data. *Journal of the American Statistical Association*, 96(453):185–193, 2001.
- [167] Kan Xu and Hamsa Bastani. Learning across bandits in high dimension via robust statistics. *arXiv preprint arXiv:2112.14233*, 2021.

- [168] Heng Yang and Marco Pavone. Object pose estimation with statistical guarantees: Conformal keypoint detection and geometric uncertainty propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8947–8958, 2023.
- [169] Tao Yin, Chenzhengyi Liu, Fangyu Ding, Ziming Feng, Bo Yuan, and Ning Zhang. Graph-based stock correlation and prediction for high-frequency trading systems. *Pattern Recognition*, 122:108209, 2022.
- [170] Donald G York, J Adelman, John E Anderson Jr, Scott F Anderson, James Annis, Neta A Bahcall, JA Bakken, Robert Barkhouser, Steven Bastian, Eileen Berman, et al. The sloan digital sky survey: Technical summary. *The Astronomical Journal*, 120(3):1579, 2000.
- [171] Menggang Yu and Bin Nan. A revisit of semiparametric regression models with missing data. *Statistica Sinica*, pages 1193–1212, 2006.
- [172] Margaux Zaffran, Olivier Féron, Yannig Goude, Julie Josse, and Aymeric Dieuleveut. Adaptive conformal predictions for time series. In *International Conference on Machine Learning*, pages 25834–25866. PMLR, 2022.
- [173] Ernst Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1929.
- [174] Anru Zhang, Lawrence D. Brown, and T. Tony Cai. Semi-supervised inference: General theory and estimation of means. *The Annals of Statistics*, 47(5):2538 – 2566, 2019.
- [175] Yuqian Zhang and Jelena Bradic. High-dimensional semi-supervised learning: in search of optimal inference of the mean. *Biometrika*, 109(2):387–403, 2022.
- [176] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.
- [177] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130, 2009.
- [178] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. *Thesis*, 2005.
- [179] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.