

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Model Aggregation: Performance Guarantees and Loss-Robust Prediction

Permalink

<https://escholarship.org/uc/item/0hh2n1qd>

ISBN

9798189270642

Author

Kim, Hyunsuk

Publication Date

2026-05-01

Peer reviewed|Thesis/dissertation

Model Aggregation: Performance Guarantees and Loss-Robust Prediction

by

Hyunsuk Kim

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael W. Mahoney, Co-chair

Professor Ryan J. Tibshirani, Co-chair

Professor Adityanand Guntuboyina

Professor William Fithian

Spring 2026

Model Aggregation: Performance Guarantees and Loss-Robust Prediction

Copyright 2026

by

Hyunsuk Kim

Abstract

Model Aggregation: Performance Guarantees and Loss-Robust Prediction

by

Hyunsuk Kim

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Michael W. Mahoney, Co-chair

Professor Ryan J. Tibshirani, Co-chair

Model aggregation, or ensembling, is a fundamental technique in modern machine learning for improving predictive performance. This dissertation develops a theoretical understanding of the effectiveness of model aggregation through two complementary perspectives: performance and robustness.

The first part of the dissertation focuses on the performance of aggregation in classification tasks. We analyze the majority vote classifier and provide an understanding of when and how much aggregation improves over individual models. We introduce a mild condition, competence, under which ensembling does not hurt. We further introduce two quantities: the disagreement error ratio (DER) and polarization, to establish sharp bounds on the performance of the majority vote classifier. Using the sharp bound, we characterize its asymptotic behavior and demonstrate that ensemble performance can be estimated from a small number of models.

The second part of the dissertation studies how to aggregate models in a way that is robust across a range of evaluation criteria. We consider the quantile forecasting problem in an online setting. We construct an aggregation algorithm that achieves $\tilde{O}(\sqrt{(\log B)/T})$ regret simultaneously over *all* proper losses that satisfy mild regularity conditions, where B is the number of base forecasters. Here, proper losses are those whose expectation is minimized by predicting the true quantiles of the target. The proposed algorithm is computationally efficient, produces coherent predictions without quantile crossing, and exhibits strong empirical performance, particularly in challenging scenarios where base models perform poorly.

Contents

Contents	i
1 Introduction	1
1.1 Overview of contributions.	2
2 Understanding When Majority Vote Classifiers Are Effective	4
2.1 Introduction	4
2.2 Background and preliminaries	6
2.3 Ensemble improvement, competence, and disagreement-error ratio	8
2.4 Evaluating the theory	11
2.5 Ensemble improvement is low in the interpolating regime	12
2.6 Discussion and conclusion	16
3 Performance Guarantees for Majority Vote Classifiers	17
3.1 Introduction	17
3.2 Background and preliminaries	19
3.3 The Polarization of an ensemble	20
3.4 Entropy-restricted ensembles	24
3.5 A universal law for ensembling	26
3.6 Discussion and conclusion	29
4 Loss-Robust Quantile Prediction for Proper Losses	31
4.1 Introduction	31
4.2 Background and preliminaries	32
4.3 Quantile omniprediction for proper losses	36
4.4 Evaluating the theory	40
4.5 Discussion and conclusion	44
Bibliography	45
A Appendix to Chapter 2	53
A.1 Proofs of main results	53
A.2 Additional empirical results	59

A.3	Pathological ensembles where majority vote fails	61
B	Appendix to Chapter 3	64
B.1	More discussion on competence	64
B.2	Proofs of main results	67
B.3	Experiment details	73
C	Appendix to Chapter 4	75
C.1	Proofs of main results	75
C.2	Single quantile omniprediction for proper losses	84
C.3	Quantile omniprediction for weighted quantile losses	88
C.4	Additional algorithmic details	95
C.5	Additional empirical results	99
C.6	Miscellaneous additional details	102

Acknowledgments

I have been extraordinarily fortunate to be surrounded by people who helped my PhD journey in many different ways. This work would not have been possible without their guidance and unwavering support.

First and foremost, I would like to express my deepest gratitude to my advisors, Michael Mahoney and Ryan Tibshirani. Michael, despite his extremely busy schedule, always made time for thoughtful and detailed discussions. During those meetings, I often came with partially formed or unorganized ideas, but left with clarity and a clear direction. His ability to connect ideas at a high level and view problems from a broader perspective has profoundly shaped how I think about research.

Ryan has also played a central role in shaping how I think about research. Working with him helped me develop a deeper curiosity about research. Every meeting with him opened new directions to explore and left me with questions that pushed me to think about them in more depth. I also greatly appreciate his care and effort in building a close and supportive research group. It has been a privilege and a joy to be part of that community.

I have also been very fortunate to have wonderful collaborators and mentors throughout my PhD. My research journey began with regular interactions with Ryan Theisen, Liam Hodgkinson, and Yaoqing Yang, who were always generous with their time and advice and guided me patiently through every stage of research. I learned how to approach research problems, develop ideas, and write with clarity. I would also like to thank Isaac for his inspiration and thoughtful guidance on my final project, which I really enjoyed, as well as working with him.

Beyond research, I have been incredibly lucky to have amazing friends. When I first moved to the United States, Taejoo Ahn and Karissa Huang greatly helped me adjust to life in a new environment. My friends in the department, Seunghoon Paik, Tiffany Ding, Sizhu Lu, Andy Shen, Austin Zane, Jeremy Goldwasser and Selma Mazioud filled my time in Evans Hall with warmth and joy. In the later years of my PhD, I especially appreciate my friends Jihoon Choi and Jeonghoon Kim, with whom I shared both the joys and challenges of the PhD journey, always supporting and encouraging each other. I would also like to thank my long time friends, Seongwon Yoon, Eunseok Lee, Hyungyu Park and Duhyun Kim, who were always there to listen, share, and support me through every phase of this journey.

I would like to extend my appreciation to La Shana and the staff of the UC Berkeley Statistics Department for their efficiency and kindness. Their prompt support behind the scenes made many aspects of my PhD life much easier.

Finally, and most importantly, I thank my family. Their unconditional support has been the foundation that made everything possible. I am grateful to my parents and my younger sister for their constant encouragement from Korea. I also thank my aunt and uncle in Irvine, who supported me since my very first day in the United States. Without my family, I would not be who I am today.

Chapter 1

Introduction

The idea of combining multiple opinions to improve decision-making has a long intellectual history. One of the earliest formal treatments appears in Condorcet’s Jury Theorem [Con85] (1785), which shows that if each member of a voting group is more likely than not to make a correct decision ($p > 0.5$), then the majority vote is more accurate than any individual, and its accuracy increases with the size of the group. This result formalizes a central intuition: aggregating imperfect yet diverse predictors can improve overall performance.

In modern machine learning, model aggregation, more commonly referred to as ensembling, has become a standard approach for improving predictive accuracy. Beyond classical methods such as random forests [Bre01] and gradient boosting [Fri01], large-scale model combination has played a decisive role in benchmark competitions. A prominent example is the Netflix Prize [BL07], where teams progressively improved recommendation accuracy by combining a large number of models. Near the end of the competition, leading teams merged their approaches into large ensembles, and the final winning solution relied on extensive model combination rather than a single dominant method [Kor09].

In the years since, advances in computational resources have enabled a shift from post hoc aggregation toward architectures that incorporate aggregation as an internal component. In particular, modern large-scale models often integrate multiple subnetworks within and learn to combine them during training. A prominent example is mixture-of-experts (MoE) architectures, which pass inputs across specialized subnetworks and adaptively weight their outputs, thereby embedding ensembling directly into the model.

Despite its empirical success, the theoretical understanding of model aggregation remains incomplete. Two properties are commonly associated with ensembling: improved predictive accuracy and increased robustness. In regression settings with convex loss functions, these properties admit clean formalizations. Jensen’s inequality implies that the loss of an averaged predictor does not exceed the average loss of its components, and averaging typically reduces variance. However, these arguments do not extend naturally to settings such as classification with discrete decisions or regression under non-convex losses. A natural question, then, is whether model aggregation continues to improve performance in such settings. More broadly,

can model aggregation yield reliable performance across a range of loss functions?

1.1 Overview of contributions.

This dissertation develops a theoretical understanding of model aggregation from two complementary perspectives: performance and robustness.

Performance of aggregation (Chapters 2–3). We begin with the majority vote classifier, a canonical aggregation rule in classification. Unlike averaging in regression, majority vote is discrete and non-smooth, which limits the applicability of standard analytical tools. Existing results often yield bounds that are either loose or uninformative.

Chapter 2 clarifies when and how much majority vote improves upon individual classifiers. Under a mild *competence* condition (Assumption B.1), we show that aggregation cannot degrade performance. We then quantify the improvement relative to the average error of individual classifiers through the error improvement ratio (EIR, Definition 2.3). We derive bounds on EIR as linear functions of the disagreement error ratio (DER, Definition 2.4), highlighting the role of diversity in determining ensemble performance.

Chapter 3 refines this analysis by introducing *polarization* (Definition 3.1), a quantity that captures higher-order structure in the joint behavior of classifiers. Polarization measures how prediction errors are distributed across data points. Using this concept, we obtain further refined bounds on majority vote performance, characterize its asymptotic behavior as the number of classifiers grows, and propose a method for estimating ensemble performance from a small subset of models.

Loss-robust model aggregation (Chapter 4). Chapters 2 and 3 focus on performance under a fixed loss function. In contrast, Chapter 4 considers a class of loss functions. In many applications, no single loss function is universally accepted. Different evaluation criteria may lead to different notions of optimality. This motivates the framework of *omniprediction* [GKR⁺22], which seeks predictors that perform well relative to a reference class across multiple loss functions.

We develop an omniprediction framework for quantile forecasting in an online probabilistic forecasting setting. We consider *all* proper losses for quantile levels that satisfy mild regularity conditions, where proper losses (Definition 4.1) are those whose expectation is minimized by predicting the true quantiles of the target. Proper losses cover a broad class of decision problems involving asymmetric costs. Given a collection of base forecasters, we construct an aggregated predictor that minimizes regret with respect to the base forecasters simultaneously over all such loss functions.

We propose a computationally efficient algorithm (Algorithm 1) based on a two-player game formulation that achieves $\tilde{O}(\sqrt{(\log B)/T})$ regret guarantee, where B is the number of base forecasters (Theorem 4.3). The algorithm always outputs coherent (sorted) multi-quantile

predictions even if base forecaster predictions are not coherent. Additionally, empirical results show that our algorithm performs particularly well compared to other algorithms when base forecaster predictions incur large errors.

Chapter 2

Understanding When Majority Vote Classifiers Are Effective

This chapter is based on the paper "When are ensembles really effective?", published in NeurIPS 2023, co-authored with Ryan Theisen, Yaoqing Yang, Liam Hodgkinson, and Michael W. Mahoney [TKY⁺24].

2.1 Introduction

The fundamental ideas underlying ensemble methods can be traced back at least two centuries, with Condorcet’s Jury Theorem among its earliest developments [Con85]. This result asserts that if each juror on a jury makes a correct decision independently and with the same probability $p > 1/2$, then the majority decision of the jury is more likely to be correct with each additional juror. The general principle of aggregating knowledge across imperfectly correlated sources is intuitive, and it has motivated many ensemble methods used in modern statistics and machine learning practice. Among these, tree-based methods like random forests [Bre01] and XGBoost [CG16] are some of the most effective and widely-used.

With the growing popularity of deep learning, a number of approaches have been proposed for ensembling neural networks. Perhaps the simplest of them are so-called deep ensembles, which are ensembles of neural networks trained from independent initializations [ABP⁺22, ABPC22, FHL19]. In some cases, it has been claimed that such deep ensembles provide significant improvement in performance [FHL19, OFR⁺19, ALMV20]. Such ensembles have also been used to obtain uncertainty estimates for prediction [LPB17], and to provide more robust predictions under distributional shift. However, the benefits of deep ensembling are not universally accepted. Indeed, other works have found that ensembling is less necessary for larger models, and that in some cases a single larger model can perform as well as an ensemble of smaller models [HVD⁺15, BCNM06, GJS⁺20, ABP⁺22]. Similarly mixed results, where empirical performance does not conform with expectations, have been reported in the Bayesian approach to deep learning [INLW21]. Furthermore, an often-cited practical

issue with ensembling, especially of large neural networks, is the constraint of storing and performing inference with many distinct models.

In light of the increase in computational cost, it is of great value to understand exactly when we might expect ensembling to improve performance substantially. In particular, consider the following practical scenario: a practitioner has trained a single (perhaps large and expensive) model, and would like to know whether they can expect significant gains in performance from training additional models and ensembling them. This question lacks a sufficient answer, both from the theoretical and empirical perspectives, and hence motivates the main question of the present work:

When are ensembles *really* effective?

The present work addresses this question, both theoretically and empirically, under very general conditions. We focus our study on the most popular ensemble classifier—the *majority vote classifier* (Definition 2.1), which we denote by h_{MV} —although our framework also covers variants such as weighted majority vote methods.

Theoretical results. Our main theoretical contributions, contained in Section 2.3, are as follows. First, we formally define the *ensemble improvement rate* (EIR, Definition 2.3), which measures the decrease in error rate from ensembling, on a relative scale. We then introduce a new condition called *competence* (Assumption 2.1) that rules out pathological cases, allowing us to prove stronger bounds on the ensemble improvement rate. Specifically, we prove that 1) competent ensembles can never hurt performance (Theorem 2.1), and 2) in Theorem 2.2 we show that the EIR can be upper and lower bounded by linear functions of the *disagreement-error ratio* (DER, Definition 2.4). Our theoretical results predict that ensemble improvement will be high whenever disagreement is large relative to the average error rate (i.e. $\text{DER} > 1$). Moreover, in Appendix A.1.3, we show that as Corollaries of our theoretical results, we obtain new bounds on the error rate of the majority vote classifier that significantly improve on previous results, provided the competence assumption is satisfied.

Empirical results. In light of our new theoretical understanding of ensembling, we perform a detailed empirical analysis of ensembling in practice. In Section 2.4, we evaluate the assumptions and predictions made by the theory presented in Section 2.3. In particular, we verify on a variety of tasks that the competence condition holds, and we verify empirically the linear relationship between the EIR and the DER, as predicted by our bounds, and suggest directions through which our theoretical results might be improved. In Section 2.5, we provide significant evidence for distinct behavior arising for ensembles in and out of the *interpolating regime*, i.e., when each of the constituent classifiers in an ensemble has sufficient capacity to achieve zero training error. We show 1) that interpolating ensembles exhibit consistently lower ensemble improvement rates, and 2) that this corresponds to ensembles transitioning (sometimes sharply) from the regime $\text{DER} > 1$ to $\text{DER} < 1$. Finally, we also

show that tree-based ensembles represent a unique exception to this phenomenon, making them particularly well-suited to ensembling.

In addition to the results presented in the main text, we provide supplemental theoretical results (including all proofs) in Appendix A.1, as well as supplemental empirical results in Appendix A.2.

2.2 Background and preliminaries

In this section, we present some background as well as preliminary results of independent interest that set the context for our main theoretical and empirical results.

2.2.1 Setup

In this work, we focus on the K -class classification setting, wherein the data $(X, Y) \in \mathcal{X} \times \mathcal{Y} \sim \mathcal{D}$ consist of features $\mathbf{x} \in \mathcal{X}$ and labels $y \in \mathcal{Y} = \{1, \dots, K\}$. Classifiers are then functions $h : \mathcal{X} \rightarrow \mathcal{Y}$ that belong to some set $\mathcal{H}$. To measure the performance of a single classifier h on the data distribution $\mathcal{D}$, we use the usual error rate:

$$L_{\mathcal{D}}(h) = \mathbb{E}_{X, Y \sim \mathcal{D}}[\mathbb{1}(h(X) \neq Y)].$$

For notational convenience, we drop the explicit dependence on $\mathcal{D}$ whenever it is apparent from context.

A central object in our study is a distribution ρ over classifiers. Depending on the context, this distribution could represent a variety of different things. For example, ρ could be: i) A discrete distribution on a finite set of classifiers $\{h_1, \dots, h_M\}$ with weights $\rho_1, \dots, \rho_M$, e.g., representing normalized weights in a weighted ensembling scheme; ii) A distribution over parameters θ of a parametric family of models, h_{θ} , determined, e.g., by a stochastic optimization algorithm with random initialization; iii) A Bayesian posterior distribution.

The distribution ρ induces two error rates of interest. The first is the **average error rate of any single classifier** under ρ , defined to be $\mathbb{E}_{h \sim \rho}[L(h)]$. The second is the **error rate of the majority vote classifier**, h_{MV} , which is defined for a distribution ρ as follows.

Definition 2.1 (Majority vote classifier). *Given ρ , the **majority vote classifier** is the classifier which, for an input $\mathbf{x}$, predicts the most probable class for this input among classifiers drawn from ρ ,*

$$h_{\text{MV}}(\mathbf{x}) = \arg \max_j \mathbb{E}_{h \sim \rho}[\mathbb{1}(h(\mathbf{x}) = j)].$$

In the Bayesian context, $\rho = \rho(h \mid X_{\text{train}}, y_{\text{train}})$ is a posterior distribution over classifiers. In this case, the majority vote classifier is often called the Bayes classifier, and the error rate

$L(h_{\text{MV}})$ is referred to as the Bayes error rate. In such contexts, the average error rate is often referred to as the Gibbs error rate associated with ρ and $\mathcal{D}$.

Finally, we will present results in terms of the *disagreement rate* between classifiers drawn from a distribution ρ , defined as follows.

Definition 2.2 (Disagreement). *The **disagreement rate** between two classifiers h, h' is given by $D_{\mathcal{D}}(h, h') = \mathbb{E}_{X \sim \mathcal{D}}[\mathbb{1}(h(X) \neq h'(X))]$. The **expected disagreement rate** is $\mathbb{E}_{h, h' \sim \rho}[D_{\mathcal{D}}(h, h')]$, where $h, h' \sim \rho$ are drawn independently.*

2.2.2 Prior work

Ensembling theory. Perhaps the simplest general relation between the majority vote error rate and the average error rate guarantees only that the majority vote classifier is no worse than twice the average error rate [LMRR17, MLIS20]. To see this, let $W_{\rho} \equiv W_{\rho}(X, Y) = \mathbb{E}_{h \sim \rho}[\mathbb{1}(h(X) \neq Y)]$ denote the proportion of erroneous classifiers in the ensemble for a randomly sampled input-output pair $(X, Y) \sim \mathcal{D}$, and note that $\mathbb{E}[W_{\rho}] = \mathbb{E}[L(h)]$. Then, by a “first-order” application of Markov’s inequality, we have that

$$0 \leq L(h_{\text{MV}}) \leq \mathbb{P}(W_{\rho} \geq 1/2) \leq 2 \mathbb{E}[W_{\rho}] = 2 \mathbb{E}_{h \sim \rho}[L(h)]. \quad (2.1)$$

This bound is almost always uninformative in practice. Indeed, it may seem surprising that an ensemble classifier could perform *worse* than the average of its constituent classifiers, much less a factor of two worse. Nonetheless, the first-order upper bound is, in fact, tight: there exist distributions ρ (over classifiers) and $\mathcal{D}$ (over data) such that the majority vote classifier is twice as erroneous as any one classifier, on average. As one might expect, however, this tends to happen only in pathological cases; we give examples of such ensembles in Appendix A.3.

To circumvent the shortcomings of the simple first-order bound, more recent approaches have developed bounds incorporating “second-order” information from the distribution ρ [MLIS20]. One successful example of this is given by a class of results known as C-bounds [GLL⁺15, LMRR17]. The most general form of these bounds states, provided $\mathbb{E}[M_{\rho}(X, Y)] > 0$, that $L(h_{\text{MV}}) \leq 1 - \mathbb{E}[M_{\rho}(X, Y)]^2 / \mathbb{E}[M_{\rho}^2(X, Y)]$, where $M_{\rho}(X, Y) = \mathbb{E}_{h \sim \rho}[\mathbb{1}(h(X) = Y)] - \max_{j \neq Y} \mathbb{E}_{h \sim \rho}[\mathbb{1}(h(X) = j)]$ is called the *margin*. In the binary classification case, the condition $\mathbb{E}[M_{\rho}(X, Y)] > 0$ is equivalent to the assumption $\mathbb{E}_{h \sim \rho}[L(h)] < 1/2$. Hence, it can be viewed as a requirement that individual classifiers be “weak learners.” The same condition is used to derive a very similar bound for random forests in [Bre01], which is then further upper bounded to obtain a more intuitive (though weaker) bound in terms of the “c/s2” ratio. Relatedly, [MLIS20] obtains a bound on the error rate of the majority vote classifier, in the special case of binary classification, directly in terms of the disagreement rate, taking the form $4\mathbb{E}[L(h)] - 2\mathbb{E}[D(h, h')]$. We note that our theory improves this bound by factor of 2 (See Appendix A.1.3). Other results obtain similar expressions, but in terms of different loss functions, e.g. cross-entropy loss [ABP⁺22, OCnM22].

Other related studies. In addition to theoretical results, there have been a number of recent empirical studies investigating the use of ensembling. Perhaps the most closely related is [ABPC22], which show, perhaps surprisingly, that ensembles do not benefit significantly from encouraging diversity during training. In contrast to the present work, [ABPC22] focuses on the cross entropy loss for classification (which facilitates somewhat simpler theoretical analysis), whereas we focus on the more intuitive and commonly used classification error rate. Moreover, while [ABPC22] study the ensemble improvement *gap* (i.e., the difference between the average loss of a single classifier and the ensemble loss), we focus on the gap in error rates on a relative scale. As we show, this provides *much* finer insights into ensemble improvement. To complement this, [FHL19] study ensembling from a loss landscape perspective, evaluating how different approaches to ensembling, such as deep ensembles, Bayesian ensembles, and local methods like Gaussian subspace sampling compare in function and weight space diversity. Other recent work has studied the use of ensembling to provide uncertainty estimates for prediction [LPB17], and to improve robustness to out-of-distribution data [OFR⁺19], although the ubiquity of these findings has recently been questioned in [ABP⁺22].

2.3 Ensemble improvement, competence, and disagreement-error ratio

In this section, our goal is to characterize theoretically the rate at which ensembling improves performance. To do this, we first need to formalize a metric to quantify the benefit from ensembling. One natural way of measuring this improvement would be to compute the gap $\mathbb{E}_{h \sim \rho}[L(h)] - L(h_{\text{MV}})$. A similar gap was the focus of [ABPC22], although in terms of the cross-entropy loss, rather than the classification error rate. However, the unnormalized gap can be misleading—in particular, it will tend to be small whenever the average error rate itself is small, thus making it impractical to compare, e.g., across tasks of varying difficulty. Instead, we work with a normalized version of the average-minus-ensemble error rate gap, where the effect of the normalization is to measure this error in a relative scale. We call this the ensemble improvement rate.

Definition 2.3 (Ensemble improvement rate). *Given distributions ρ over classifiers and $\mathcal{D}$ over data, provided that $\mathbb{E}_{h \sim \rho}[L(h)] \neq 0$, the **ensemble improvement rate (EIR)** is defined as*

$$\text{EIR} = \frac{\mathbb{E}_{h \sim \rho}[L(h)] - L(h_{\text{MV}})}{\mathbb{E}_{h \sim \rho}[L(h)]}.$$

In contrast to the unnormalized gap, the ensemble improvement rate can be large even for very easy tasks with a small average error rate. Recall the simple first-order bound on the majority vote classifier: $L(h_{\text{MV}}) \leq 2\mathbb{E}[L(h)]$. Rearranging, we deduce that $\text{EIR} \geq -1$. Unfortunately, in the absence of additional information, this first-order bound is in fact tight: one can construct ensembles for this $L(h_{\text{MV}}) = 2\mathbb{E}[L(h)]$ (see Appendix A.3). However, this

bound is inconsistent with how ensembles generally behave in practice, and indeed it tells us nothing about when ensembling can improve performance. In the subsequent sections, we derive improved bounds on the EIR that can provide such insight.

2.3.1 Competent ensembles never hurt

Surprisingly, to our knowledge, there is no known characterization of the majority vote error rate that *guarantees* it can be no worse than the error rate of any individual classifier, on average. Indeed, it turns out this is the result of strange behavior that can arise for particularly pathological ensembles rarely encountered in practice (see Appendix A.3 for a more detailed discussion of this). To eliminate these cases, we introduce a mild condition that we call *competence*.

Assumption 2.1 (Competence). *Let $W_\rho \equiv W_\rho(X, Y) = \mathbb{E}_{h \sim \rho}[\mathbb{1}(h(X) \neq Y)]$. The ensemble ρ is **competent** if for every $0 \leq t \leq 1/2$,*

$$\mathbb{P}(W_\rho \in [t, 1/2)) \geq \mathbb{P}(W_\rho \in [1/2, 1 - t]).$$

The competence assumption guarantees that the ensemble is not pathologically bad, and in particular it eliminates the scenarios under which the naive first-order bound (2.1) is tight. As we will show in Section 2.4, the competence condition is quite mild, and it holds broadly in practice. Our first result uses competence to improve non-trivially the naive first-order bound.

Theorem 2.1. *Competent ensembles never hurt performance, i.e., $\text{EIR} \geq 0$.*

Translated into a bound on the majority vote classifier, Theorem 2.1 guarantees that $L(h_{\text{MV}}) \leq \mathbb{E}[L(h)]$, improving on the naive first-order bound (2.1) by a factor of two. To the best of our knowledge, the competence condition is the first of its kind, in that it guarantees what is widely observed in practice, i.e., that ensembling cannot *hurt* performance. However, it is insufficient to answer the question of *how much* ensembling can improve performance. To address this question, we turn to a "second-order" analysis involving the disagreement rate.

2.3.2 Quantifying ensemble improvement with the disagreement-error ratio

Our central result in this section will be to relate the EIR to the ratio of the disagreement to average error rate, which we define formally below.

Definition 2.4 (Disagreement-error ratio). *Given the distributions ρ over classifiers and $\mathcal{D}$ over data, provided that $\mathbb{E}_{h \sim \rho}[L(h)] \neq 0$, the **disagreement-error ratio (DER)** is defined as*

$$\text{DER} = \frac{\mathbb{E}_{h, h' \sim \rho}[D(h, h')]}{\mathbb{E}_{h \sim \rho}[L(h)]}.$$

Our next result relates the EIR to a linear function of the DER.

Theorem 2.2. *For any competent ensemble ρ of K -class classifiers, provided $\mathbb{E}_{h \sim \rho}[L(h)] \neq 0$, the ensemble improvement rate satisfies*

$$\text{DER} \geq \text{EIR} \geq \frac{2(K-1)}{K} \text{DER} - \frac{3K-4}{K}.$$

Note that neither Theorem 2.1 nor Theorem 2.2 is uniformly stronger. In particular, if $\text{DER} < (3K-4)/(2K-2)$ then the lower bound provided in Theorem 2.1 will be superior to the one in Theorem 2.2.

Theorem 2.2 predicts that the EIR is fundamentally governed by a linear relationship with the DER — a result that we will verify empirically in Section 2.4. Importantly, we note that there are two distinct regimes in which the bounds in Theorem 2.2 provide non-trivial guarantees.

DER small (< 1). In this case, by the trivial bound (2.1), $\text{EIR} \leq 1$, and thus the upper bound in Theorem 2.2 guarantees ensemble improvement cannot be too large whenever $\text{DER} < 1$, that is, whenever disagreement is small relative to the average error rate.

DER large (> 1). In this case, the lower bound in Theorem 2.2 guarantees ensemble improvement whenever disagreement is sufficiently large relative to the average error rate, in particular when $\text{DER} \geq (3K-4)/(2K-2) \geq 1$.

In our empirical evaluations, we will see that these two regimes ($\text{DER} > 1$ and $\text{DER} < 1$) strongly distinguish between situations in which ensemble improvement is high, and when the benefits of ensembling are significantly less pronounced.

Moreover, Theorem 2.2 captures an important subtlety in the relationship between ensemble improvement and predictive diversity. In particular, while general intuition—and a significant body of prior literature, as discussed in Section 2.2—suggests that higher disagreement leads to high ensemble improvement, this may *not* be the case if the disagreement is nominally large, but small relative to the average error rate.

Remark 2.1 (Corollaries of Theorems 2.1 and 2.2). *Using some basic algebra, the upper and lower bounds presented in Theorems 2.1 and 2.2 can easily be translated into upper and lower bounds on the error rate of the majority vote classifier itself. For the sake of space, we defer discussion of these Corollaries to Appendix A.1.3, although we note that the resulting bounds constitute significant improvements on existing bounds, which we verify both analytically (when possible) and empirically.*

Remark 2.2 (Relation to the regression case). *We remark that the bounds in Theorem 2.2 share a direct (and simpler) analogue in the regression case with the squared error loss. In*

this case, a bias-variance decomposition (taken over randomness arising from the sampling of a regressor $f \sim \rho$) yields the identity:

$$\mathbb{E}_{f \sim \rho}[L(f)] = L(\bar{f}) + \mathbb{E}_{X \sim \mathcal{D}}[\text{Var}_{f \sim \rho}[f(X)]]$$

where here the ensemble regressor is $\bar{f}(\mathbf{x}) = \mathbb{E}_{f \sim \rho}[f(\mathbf{x})]$ and $\text{Var}[\cdot]$ is the variance. Normalizing, we obtain the identity

$$\text{EIR} = \frac{\mathbb{E}_{f \sim \rho}[L(f)] - L(\bar{f})}{\mathbb{E}_{f \sim \rho}[L(f)]} = \frac{\mathbb{E}_{X \sim \mathcal{D}}[\text{Var}_{f \sim \rho}[f(X)]]}{\mathbb{E}_{f \sim \rho}[L(f)]}$$

where the latter variance-average error ratio is a natural analogue to the DER. Note also since the variance is non-negative, this expression immediately implies that $\text{EIR} \geq 0$ in the regression case. See Appendix A.1.4 for a derivation.

2.4 Evaluating the theory

In this section, we investigate the assumptions and predictions of the theory proposed in Section 2.3. In particular we will show 1) that the competence assumption holds broadly in practice, across a range of architectures, ensembling methods and datasets, and 2) that the EIR exhibits a close linear relationship with the DER, as predicted by Theorem 2.2.

Before presenting our findings, we first briefly describe the experimental settings analyzed in the remainder of the chapter. Our goal is to select a sufficiently broad range of tasks and methods so as to demonstrate the generality of our conclusions.

2.4.1 Setup for empirical evaluations

In Table 2.1 we provide a brief description of our experimental setup; more extensive experimental details can be found in Appendix A.2.1.

Table 2.1: Datasets and ensembles used in empirical evaluations, where C denotes the number of classes, and M denotes the number of classifiers.

<i>Datasets</i>			<i>Ensembles</i>			
Dataset	C	Reference	Base classifier	Ensembling	M	Reference
MNIST-5K	10	[Den12]	ResNet20-Swish	Bayesian Ens.	100	[IVHW21]
CIFAR-10	10	[KH ⁺ 09]	ResNet18	Deep Ens.	5	[KH ⁺ 09]
IMDB	2	[MDP ⁺ 11]	CNN-LSTM	Bayesian Ens.	100	[IVHW21]
QSAR	2	[BGCT19]	BERT (fine-tune)	Deep Ens.	25	[DCLT19, SYT ⁺ 22]
Thyroid	2	[QCHL87]	Random Features	Bagging	30	N/A
GLUE (7 tasks)	2-3	[WSM ⁺ 19]	Decision Trees	Random Forests	100	[PVG ⁺ 11, Bre01]

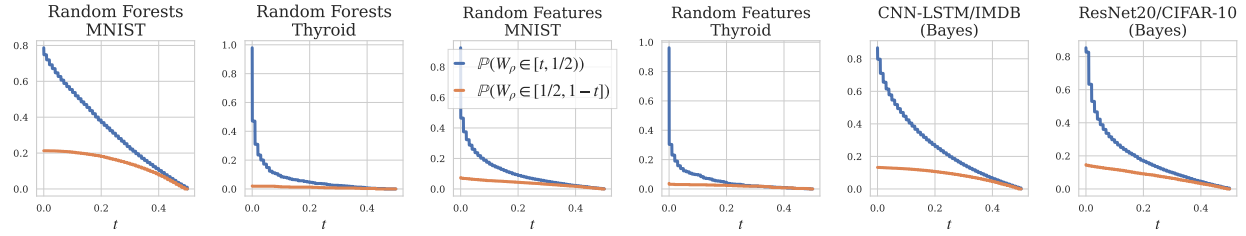


Figure 2.1: **Verifying the competence assumption in practice.** $W_\rho(X, Y)$ in Assumption 2.1 is estimated using hold-out data. Across all tasks, ■ $>$ ■, supporting Assumption 2.1.

2.4.2 Verifying competence in practice.

Our theoretical results in Section 2.3 relied on the competence condition. One might wonder whether competent ensembles exist, and if so how ubiquitous they are. Here, we test that assumption. (That is, we test not just the predictions of our theory, but also the assumptions of our theory.)

We have observed that the competence assumption is empirically very mild, and that in practice it applies very broadly. In Figure 2.1, we estimate both $\mathbb{P}(W_\rho \in [t, 1/2))$ and $\mathbb{P}(W_\rho \in [1/2, 1 - t])$ on test data, validating that competence holds for various types of ensembles across a subset of tasks. To do this, given a test set of examples $\{(\mathbf{x}_j, y_j)\}_{j=1}^m$ and classifiers $h_1, \dots, h_N$ drawn from ρ , we construct the estimator

$$\widehat{W}_\rho^{(j)} = \frac{1}{N} \sum_{n=1}^N \mathbb{1}(h_n(\mathbf{x}_j) \neq y_j),$$

and we calculate $\mathbb{P}(W_\rho \in [t, 1/2))$ and $\mathbb{P}(W_\rho \in [1/2, 1 - t])$ from the empirical CDF of $\{\widehat{W}_\rho^{(j)}\}_{j=1}^m$. In Appendix A.2, we provide additional examples of competence plots across more experimental settings (and we observe substantially the same results).

2.5 Ensemble improvement is low in the interpolating regime

In this section, we will show that the DER behaves qualitatively differently for interpolating versus non-interpolating ensembles, in particular exhibiting behavior associated with phase transitions. Such phase transitions are well-known in the statistical mechanics approach to learning [EdB01, MM17, TKM21, YHT⁺21], but they have been viewed as surprising from the more traditional approach to statistical learning theory [BHMM19]. We will use this to understand when ensembling is and is not effective for deep ensembles, and to explain why tree-based methods seem to benefit so much from ensembling across all settings. We say that a model is *interpolating* if it achieves exactly zero training error, and we say that it

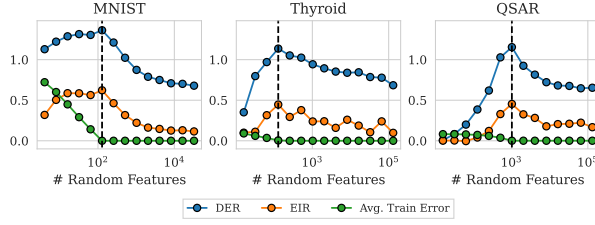


Figure 2.2: **Bagged random feature classifiers.** Black dashed line represents the interpolation threshold. Across all tasks, DER and EIR are maximized at this point, and then decrease thereafter.

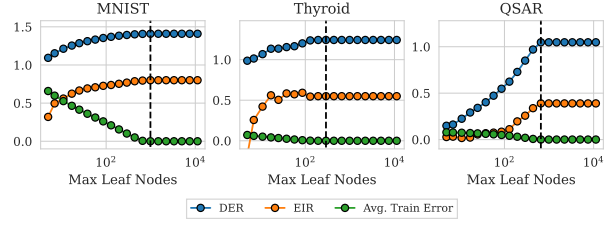
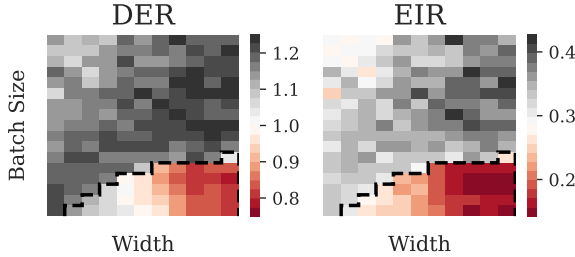
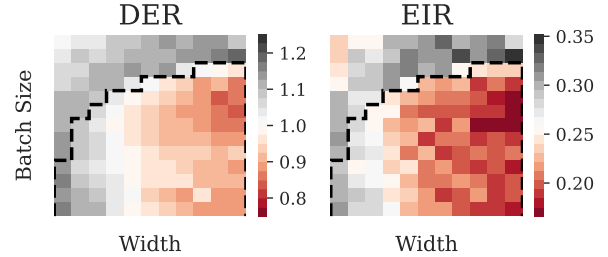


Figure 2.3: **Random forest classifiers.** Black dashed line represents the interpolation threshold. Across all tasks, DER and EIR are maximized at this point, and then remain constant thereafter.



(a) **Without LR decay.**



(b) **With LR decay.**

Figure 2.4: **Large scale studies of deep ensembles on ResNet18/CIFAR-10.** We plot the DER and EIR across a range of hyper-parameters, for two training settings: one with learning rate decay, and one without. The black dashed line indicates the *interpolation threshold*, i.e., the curve below which individual models achieve exactly zero training error. Observe that interpolating ensembles attain distinctly lower EIR than non-interpolating ensembles, and correspondingly have low DER (< 1), compared to non-interpolating ensembles with high DER (> 1).

is *non-interpolating* otherwise; we call an ensemble interpolating if each of its constituent classifiers is interpolating. Note that for methods that involve resampling of the training data (e.g., bagging methods), we define the training error as the “in-bag” training, i.e., the error evaluated only on the points a classifier was trained on.

Interpolating random feature classifiers. We first look at the bagged random feature ensembles on the MNIST, Thyroid, and QSAR datasets. In Figure 2.2, we plot the EIR, DER and training error for each of these ensembles. We observe the same phenomenon across these three tasks: as a function of model capacity, the EIR and DER are both maximized at the

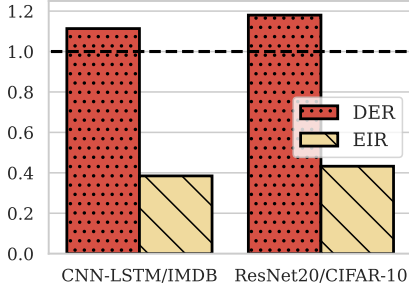


Figure 2.5: **Bayesian ensembles on IMDB and CIFAR-10.**

These provide examples of ensembles which do *not* interpolate the training data, and which have high DER (> 1) and correspondingly high EIR.

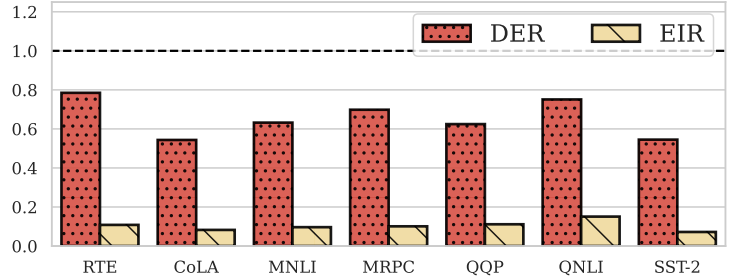


Figure 2.6: **Ensembles of fine-tuned BERT on GLUE tasks.** Here the models are large relative to the dataset size, and consequently they exhibit low DER (< 1) and EIR across all tasks.

interpolation threshold, before decreasing thereafter. This indicates that much higher-capacity models, those with the ability to easily interpolate the training data, benefit significantly less from ensembling. In particular, observe that for sufficiently high-capacity ensembles, the DER become less than 1, entering the regime in which our theory guarantees low ensemble improvement.

Interpolating deep ensembles. We next consider the DER and EIR for large-scale empirical evaluations on ResNet18/CIFAR-10 models in batch size/width space, both with and without learning rate decay. See Figure 2.4 for the results. Note that the use of learning rate decay facilitates easier interpolation of the training data during training, hence broadening the range of hyper-parameters for which interpolation occurs. The figures are colored so that ensembles in the regime $\text{DER} < 1$ are in red, while ensembles with $\text{DER} > 1$ are in grey. The black dashed line indicates the interpolation threshold, i.e., the curve in hyper-parameter space below which ensembles achieve zero training error (meaning every classifier in the ensemble has zero training error). *Observe in particular that, across all settings, all models in the regime $\text{DER} < 1$ are interpolating models; and, more generally, that interpolating models tend to exhibit much smaller DER than non-interpolating models.* Observe moreover that there can be a sharp transition between these two regimes, wherein the DER is large just at the interpolation threshold, and then it quickly decreases beyond that threshold. Correspondingly, ensemble improvement is *much* less pronounced for interpolating ensembles versus more traditional non-interpolating ensembles. This is consistent with results previously observed in the literature, e.g., [GJS⁺20]. We remark also that the behavior observed in Figure 2.4 exhibits the same phases identified in [YHT⁺21] (an example of phase transitions in learning more generally [EdB01, MM17]), although the DER itself was not considered in that previous study.

Bayesian neural networks. Next, we show that Bayesian neural networks benefit significantly from ensembling. In Figure 2.5, we plot the DER and EIR for Bayesian ensembles on the CIFAR-10 and IMDB tasks, using the ResNet20 and CNN-LSTM architectures, respectively. The samples we present here are provided by [INLW21], who use Hamiltonian Monte Carlo to sample accurately from the posterior distribution over models. For both tasks, we observe that both ensembles exhibit $\text{DER} > 1$ and high EIR. In light of our findings regarding ensemble improvement and interpolation, we note that the Bayesian ensembles by design do *not* interpolate the training data (when drawn at non-zero temperature), as the samples are drawn from a distribution not concentrated only on the modes of the training loss. While we do not perform additional experiments with Bayesian neural networks in the present work, evaluating the DER/EIR as a function of posterior temperature for these models is an interesting direction for future work. We hypothesize that the qualitative effective of decreasing the sampling temperature will be similar to that of increasing the batch size in the plots in Figure 2.4.

Fine-tuned BERT ensembles. Here, we present results for ensembles of BERT models fine-tuned on the GLUE classification tasks. These provide examples of a very large model trained on small datasets, on which interpolation is easily possible. For these experiments, we use 25 BERT models pre-trained from independent initializations provided in [SYT⁺22]. Each of these 25 models is then fine-tuned on the 7 classification tasks in the GLUE [WSM⁺19] benchmark set and evaluated on relatively small test sets, ranging in size from 250 (RTE) to 40,000 (QQP) samples. In Figure 2.6, we plot the EIR and DER across these benchmark tasks and observe that, as predicted, the ensemble improvement rate is low and DER is uniformly low (< 1).

The unique case of random forests. Random forests are one of the most widely-used ensembling methods in practice. Here, we show that the effectiveness of ensembling is much greater for random forest models than for highly-parameterized models like the random feature classifiers and the deep ensembles. Note that for random forests, interpolation of the training data *is* possible, in particular whenever the number of terminal leaf nodes is sufficiently large (where here we again compute the average training error using on the in-bag training examples for each tree), but it is not possible to go “into” the interpolating regime. In Figure 2.3, we plot the DER, EIR, and training error as a function of the max number of leaf nodes (a measure of model complexity). Before the interpolation threshold, both the EIR and DER increase as a function of model capacity, in line with what is observed for the random feature and deep ensembles. However, we observe distinct behavior at the interpolation threshold: both EIR and DER become constant past this threshold. This is fundamental to tree-based methods, due to the method by which they are fit, e.g., using a standard procedure like CART [BFSO84]. As soon as a tree achieves zero training error, any impurity method used to split the nodes further is saturated at zero, and therefore the models cannot continue to grow. This indicates that trees are particularly well-suited to ensembling

across all hyper-parameter values, in contrast to other parameterized types of classifiers.

2.6 Discussion and conclusion

To help answer the question of when ensembling is effective, we introduce the ensemble improvement rate (EIR), which we then study both theoretically and empirically. Theoretically, we provide a comprehensive characterization of the EIR in terms of the disagreement-error ratio DER. The results are based on a new, mild condition called *competence*, which we introduce to rule out pathological cases that have hampered previous theoretical results. Using a simple first-order analysis, we show that the competence condition is sufficient to guarantee that ensembling cannot hurt performance—something widely observed in practice, but surprisingly unexplained by existing theory. Using a second-order analysis, we are able to theoretically characterize the EIR, by upper and lower bounding it in terms of a linear function of the DER. On the empirical side, we first verify the assumptions of our theory (namely that the competence assumption holds broadly in practice), and we show that our bounds are indeed descriptive of ensemble improvement in practice. We then demonstrate that improvement decreases precipitously for interpolating ensembles, relative to non-interpolating ones, providing a very practical guideline for when to use ensembling.

Our work leaves many directions to explore, of which we name a few promising ones. First, while our theory represents a significant improvement on previous results, there are still directions to extend our analysis. Is it possible to refine our analysis further to incorporate information about the data and/or model architecture? Second, can we formalize the connection between ensemble effectiveness and the interpolation point, and relate it to similar ideas in the literature?

Chapter 3

Performance Guarantees for Majority Vote Classifiers

This chapter is based on the paper "How many classifiers do we need?", published in NeurIPS 2024, co-authored with Liam Hodgkinson, Ryan Theisen, and Michael W. Mahoney [KHTM24].

3.1 Introduction

As performance gains through scaling data and/or model size experience diminishing returns, it is becoming increasingly popular to turn to *ensembling*, where the predictions of multiple models are combined, both to improve accuracy and to form more robust conclusions than any individual model alone can provide. In some cases, ensembling can produce substantial benefits, particularly when increasing model size becomes prohibitive. In particular, for large neural network models, *deep ensembles* [LPB17] are especially popular. These ensembles consist of independently trained models on the same dataset, often using the same hyperparameters, but starting from different initializations.

The cost of producing new classifiers can be steep, and it is often unclear whether the additional performance gains are worth the cost. Assuming that constructing two or three classifiers is relatively cheap, procedures capable of deciding whether to continue producing more classifiers are needed. To do so requires a precise understanding of how to predict ensemble performance. Of particular interest are majority vote strategies in classification tasks, noting that regression tasks can also be formulated in this way by clustering outputs. In this case, one of the most effective avenues for predicting performance is the *disagreement* [JNBK21, BJRK22]: measuring the degree to which classifiers provide different conclusions over a given dataset. Disagreement is concrete, easy to compute, and strongly linearly correlated with majority vote prediction accuracy, leading to its use in many applications. However, *a priori*, the precise linear relationship between disagreement and accuracy is unclear, preventing the use of disagreement for predicting ensemble performance.

Our goal in this chapter is to go beyond disagreement-based analysis to provide a more quantitative understanding of the number of classifiers one should use to achieve a desired level of performance in modern practical applications, in particular for neural network models. In more detail, our contributions are as follows.

- (i) We introduce and define the concept of **polarization**, a notion that measures the higher-order dispersity of the error rates at each data point, and which indicates how polarized the ensemble is relative to the ground truth. We state and prove an upper bound for polarization (Theorem 3.1). Inspired by the theorem, we propose what we call a **neural polarization law** (Conjecture 3.1): most interpolating (Definition 3.2) neural network models are $4/3$ -polarized. We provide empirical results supporting the conjecture (Figures 3.1 and 3.2).
- (ii) Using the notion of polarization, we develop a refined set of **bounds on the majority vote test error rate**. For one, we provide a sharpened bound for any ensembles with a **finite number** of classifiers (Corollary 3.1). For the other, we offer a new, tighter bound under an additional condition on the **entropy** of the ensemble (Theorem 3.4). We provide empirical results that demonstrate our new bounds perform significantly better than the existing bounds on the majority vote test error (Figure 3.3).
- (iii) The **asymptotic behavior of the majority vote error rate** is determined as the number of classifiers increases (Theorem 3.5). Consequently, we show that we can predict the performance for a larger number of classifiers from that of a smaller number. We provide empirical results that show such predictions are accurate across various pairs of model architecture and dataset (Figure 3.4).

In Section 3.2, we define the notations that will be used throughout the chapter, and we introduce upper bounds for the error rate of the majority vote from previous work. The next three sections are the main part of this chapter. In Section 3.3, we introduce the notion of polarization, η_ρ , which plays a fundamental role in relating the majority vote error rate to average error rate and disagreement. We explore the properties of the polarization and present empirical results that corroborate our claims. In Section 3.4, we present tight upper bounds for the error rate of the majority vote for ensembles that satisfy certain conditions; and in Section 3.5, we prove how disagreement behaves in terms of the number of classifiers. All of these ingredients are put together to estimate the error rate of the majority vote for a large number of classifiers using information from only three sampled classifiers. In Section 3.6, we provide a brief discussion and conclusion. Additional material is presented in the appendices.

3.2 Background and preliminaries

In this section, we introduce notation that we use throughout the chapter, and we summarise previous work on the performance of the majority vote error rate.

3.2.1 Notations

We focus on K -class classification problems, with features $X \in \mathcal{X}$, labels $Y \in [K] = \{1, 2, \dots, K\}$ and feature-label pairs $(X, Y) \sim \mathcal{D}$. A classifier $h : \mathcal{X} \rightarrow [K]$ is a function that maps a feature to a label. We define the error rate of a single classifier h , and the disagreement and the tandem loss [MLIS20] between two classifiers, h and h' , as the following:

$$\begin{aligned} \text{Error rate : } L(h) &= \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h(X) \neq Y)] \\ \text{Disagreement : } D(h, h') &= \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h(X) \neq h'(X))] \\ \text{Tandem loss : } L(h, h') &= \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h(X) \neq Y)\mathbb{1}(h'(X) \neq Y)], \end{aligned}$$

where the expectation $\mathbb{E}_{\mathcal{D}}$ is used to denote $\mathbb{E}_{(X,Y) \sim \mathcal{D}}$. Next, we consider a distribution of classifiers, ρ , which may be viewed as an *ensemble* of classifiers. This distribution can represent a variety of different cases. Examples include: (1) a discrete distribution over finite number of h_i , e.g., a weighted sum of h_i ; and (2) a distribution over a parametric family h_{θ} , e.g., a distribution of classifiers resulting from one or multiple trained neural networks. Given the ensemble ρ , the (*weighted*) *majority vote* $h_{\rho}^{\text{MV}} : \mathcal{X} \rightarrow [K]$ is defined as

$$h_{\rho}^{\text{MV}}(x) = \arg \max_{y \in [K]} \mathbb{E}_{\rho}[\mathbb{1}(h(x) = y)].$$

Again, $\mathbb{E}_{\rho}$ denotes $\mathbb{E}_{h \sim \rho}$, and we use $\mathbb{E}_{\rho}, \mathbb{E}_{\rho^2}, \mathbb{P}_{\rho}$ for $\mathbb{E}_{h \sim \rho}, \mathbb{E}_{(h,h') \sim \rho^2}, \mathbb{P}_{h \sim \rho}$, respectively, throughout the chapter. In this sense, $\mathbb{E}_{\rho}[L(h)]$ represents *the average error rate* under a distribution of classifiers ρ and $\mathbb{E}_{\rho^2}[D(h, h')]$ represents *the average disagreement* between classifiers under ρ . Hereafter, we refer to $\mathbb{E}_{\rho}[L(h)]$, $\mathbb{E}_{\rho^2}[D(h, h')]$, and $L(h_{\rho}^{\text{MV}})$ as the **average error rate**, the **disagreement**, and the **majority vote error rate**, respectively, with

$$L(h_{\rho}^{\text{MV}}) = \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h_{\rho}^{\text{MV}}(X) \neq Y)].$$

Lastly, we define the point-wise error rate, $W_{\rho}(X, Y)$, which will serve a very important role in this chapter (for clarity, we will denote $W_{\rho}(X, Y)$ by W_{ρ} unless otherwise necessary):

$$W_{\rho}(X, Y) = \mathbb{E}_{\rho}[\mathbb{1}(h(X) \neq Y)]. \quad (3.1)$$

3.2.2 Bounds on the majority vote error rate

The simplest relationship between the majority vote error $L(h_{\rho}^{\text{MV}})$ and the average error rate $\mathbb{E}_{\rho}[L(h)]$ was introduced in [McA98]. It states that the error in the majority vote classifier cannot exceed twice the average error rate:

$$L(h_{\rho}^{\text{MV}}) \leq 2\mathbb{E}_{\rho}[L(h)] \quad (3.2)$$

A simple proof for this relationship can be found in [MLIS20] using Markov’s inequality. Although (3.2) does not provide useful information in practice, it is worth noting that this bound is, in fact, tight. There exist pathological examples where h_ρ^{MV} exhibits twice the average error rate (see Appendix C in [TKY+24]). This suggests that we can hardly obtain a useful or tighter bound by relying on only the “first-order” term, $\mathbb{E}_\rho[L(h)]$.

Accordingly, more recent work constructed bounds in terms of “second-order” quantities, $\mathbb{E}_{\rho^2}[L(h, h')]$ and $\mathbb{E}_{\rho^2}[D(h, h')]$. In particular, [LMRR17] and [MLIS20] designed a so-called *C-bound* using the Chebyshev-Cantelli inequality, establishing that, if $\mathbb{E}_\rho[L(h)] < 1/2$, then

$$L(h_\rho^{\text{MV}}) \leq \frac{\mathbb{E}_{\rho^2}[L(h, h')] - \mathbb{E}_\rho[L(h)]^2}{\mathbb{E}_{\rho^2}[L(h, h')] - \mathbb{E}_\rho[L(h)] + \frac{1}{4}}. \quad (3.3)$$

As an alternative approach, [MLIS20] incorporated the disagreement $\mathbb{E}_{\rho^2}[D(h, h')]$ into the bound as well, albeit restricted to the binary classification problem, to obtain:

$$L(h_\rho^{\text{MV}}) \leq 4\mathbb{E}_\rho[L(h)] - 2\mathbb{E}_{\rho^2}[D(h, h')]. \quad (3.4)$$

While (3.3) and (3.4) may be tighter in some cases, once again, there do exist pathological examples where this bound is as uninformative as the first-order bound (3.2). Motivated by these weak results, [TKY+24] take a new approach by restricting ρ to be a “good ensemble,” and introducing the *competence* condition (see Definition B.1 in our Appendix B.1). Informally, competent ensembles are those where it is more likely—in average across the data—that more classifiers are correct than not. Based on this notion, [TKY+24] prove that competent ensembles are guaranteed to have weighted majority vote error *smaller* than the weighted average error of individual classifiers:

$$L(h_\rho^{\text{MV}}) \leq \mathbb{E}_\rho[L(h)]. \quad (3.5)$$

That is, the majority vote classifier is always beneficial. Moreover, [TKY+24] proves that any competent ensemble ρ of K -class classifiers satisfy the following inequality.

$$L(h_\rho^{\text{MV}}) \leq \frac{4(K-1)}{K} \left(\mathbb{E}_\rho[L(h)] - \frac{1}{2}\mathbb{E}_{\rho^2}[D(h, h')] \right). \quad (3.6)$$

We defer further discussion of competence to Appendix B.1, where we introduce simple cases for which competence does *not* hold. In these cases, we show how one can overcome this issue so that the bounds (3.5) and (3.6) still hold. In particular, in Appendix B.1.3, we provide an example to show the bound (3.6) is tight.

3.3 The Polarization of an ensemble

In this section, we introduce a new quantity, η_ρ , which we refer to as the *polarization* of an ensemble ρ . First, we provide examples as to what this quantity represents and draw a

connection to previous studies. Then, we present theoretical and empirical results that show this quantity plays a fundamental role in relating the majority vote error rate to average error rate and disagreement. In Theorem 3.1, we prove an upper bound for the polarization η_ρ , which highlights a fundamental relationship between the polarization and the constant $\frac{4}{3}$. Inspired from the theorem, we propose Conjecture 3.1 which we call a *neural polarization law*. Figures 3.1 and 3.2 present empirical results on an image recognition task that corroborates the conjecture.

We start by defining the polarization of an ensemble. In essence, the polarization is an improved (smaller) coefficient on the Markov's inequality on $\mathbb{P}_\mathcal{D}(W_\rho > 0.5)$, where W_ρ is the point-wise error rate defined as equation (3.1). It measures how much the ensemble is “polarized” from the truth, with consideration of the distribution of W_ρ .

Definition 3.1 (POLARIZATION). *An ensemble ρ is η -polarized if*

$$\eta \mathbb{E}_\mathcal{D}[W_\rho^2] \geq \mathbb{P}_\mathcal{D}(W_\rho > 1/2). \quad (3.7)$$

*The **polarization** of an ensemble ρ is*

$$\eta_\rho := \frac{\mathbb{P}_\mathcal{D}(W_\rho > 1/2)}{\mathbb{E}_\mathcal{D}[W_\rho^2]}, \quad (3.8)$$

which is the smallest value of η satisfies inequality (3.7).

Note that the polarization always takes a value in $[0, 4]$, due to the positivity constraint and Markov's inequality. Also note that ensemble ρ with polarization η_ρ is η -polarized for any $\eta \geq \eta_\rho$.

To understand better what this quantity represents, consider the following examples. The first example demonstrates that polarization increases as the majority vote becomes more polarized from the truth, while the second example demonstrates how polarization increases when the constituent classifiers are more evenly split.

Example 1. Consider an ensemble ρ where 75% of classifiers output Label 1 with probability one, and the other 25% classifiers output Label 2 with probability one.

- **Case 1.** *The true label is Label 1 for the whole data.*

In this case, the majority vote in ρ results in zero error rate. The point-wise error rate W_ρ is 0.25 on the entire dataset, and thus $\mathbb{P}_\mathcal{D}(W_\rho > 0.5) = 0$. The polarization η_ρ is 0.

- **Case 2.** *The true label is Label 1 for half of the data and is Label 2 for the other half.*

In this case, the majority vote is only correct for half of the data. The point-wise error rate W_ρ is 0.25 for this half, and is 0.75 for the other half. The polarization η_ρ is $0.5/0.3125 = 1.6$.

- **Case 3.** *The true label is Label 2 for the whole data.*

In this case, the majority vote in ρ is wrong on every data point. The point-wise error rate W_ρ is 0.75 on the entire dataset and thus $\mathbb{P}_\mathcal{D}(W_\rho > 0.5) = 1$. The polarization η_ρ is $1/0.3125 = 3.2$.

Example 2. Now consider an ensemble ρ of which 51% of classifiers always output Label 1, and the other 49% classifiers always output Label 2.

- **Case 1.** The polarization η_ρ is now 0, the same as in Example 1.
- **Case 2.** The polarization η_ρ is $0.5/0.2501 \approx 2$, which is larger than 1.6 in Example 1.
- **Case 3.** The polarization η_ρ is now $1/0.2501 \approx 4$, which is larger than 3.2 in Example 1.

In addition, the following proposition draws a connection between polarization and the competence condition mentioned in Section 3.2.2. It states that the polarization of competent ensembles cannot be very large. The proof is deferred to Appendix B.1.2.

Proposition 3.1. *Competent ensembles are 2-polarized.*

Now we delve more into this new quantity. We introduce Theorem 3.1, which establishes (by means of concentration inequalities) an upper bound on the polarization η_ρ . The proof of Theorem 3.1 is deferred to Appendix B.2.1.

Theorem 3.1. *Let $\{(X_i, Y_i)\}_{i=1}^m$ be independent and identically distributed samples from $\mathcal{D}$ that are independent of an ensemble ρ . Then the polarization of the ensemble, η_ρ , satisfies*

$$\eta_\rho \leq \max \left\{ \frac{4}{3}, \left(\frac{\sqrt{\frac{3}{8m} \log \frac{1}{\delta}} + \sqrt{\frac{3}{8m} \log \frac{1}{\delta} + 4SP}}{2S} \right)^2 \right\}, \quad (3.9)$$

with probability at least $1 - \delta$, where $S = \frac{1}{m} \sum_{i=1}^m W_\rho^2(X_i, Y_i)$ and $P = \frac{1}{m} \sum_{i=1}^m \mathbb{1}(W_\rho(X_i, Y_i) > 1/2)$.

Surprisingly, in practice, $\eta_\rho = \frac{4}{3}$ appears to be a good choice for a wide variety of cases. See Figure 3.1 and Figure 3.2, which show the polarization η_ρ obtained from VGG11 [SZ14], DenseNet40 [HLVDMW17], ResNet18, ResNet50 and ResNet101 [HZRS16] trained on CIFAR-10 [KH+09] with various hyperparameters choices. The trend does not deviate even when evaluated on an out-of-distribution dataset, CIFAR-10.1 [RRSS19, TFF08]. For more details on these empirical results, see Appendix B.3.

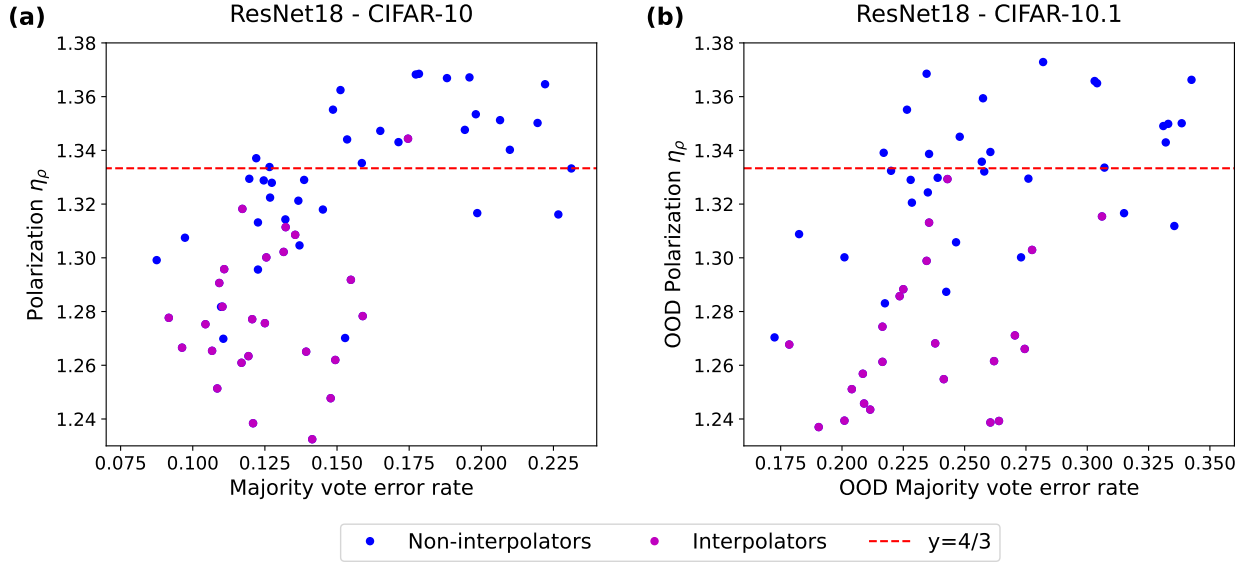


Figure 3.1: Polarizations η_ρ obtained from ResNet18 trained on CIFAR-10 with various sets of hyper-parameters tested on (a) an out-of-sample CIFAR-10 and (b) an out-of-distribution dataset, CIFAR-10.1. Red dashed line indicates $y = 4/3$, a suggested value of polarization appears in Theorem 3.1 and Conjecture 3.1.

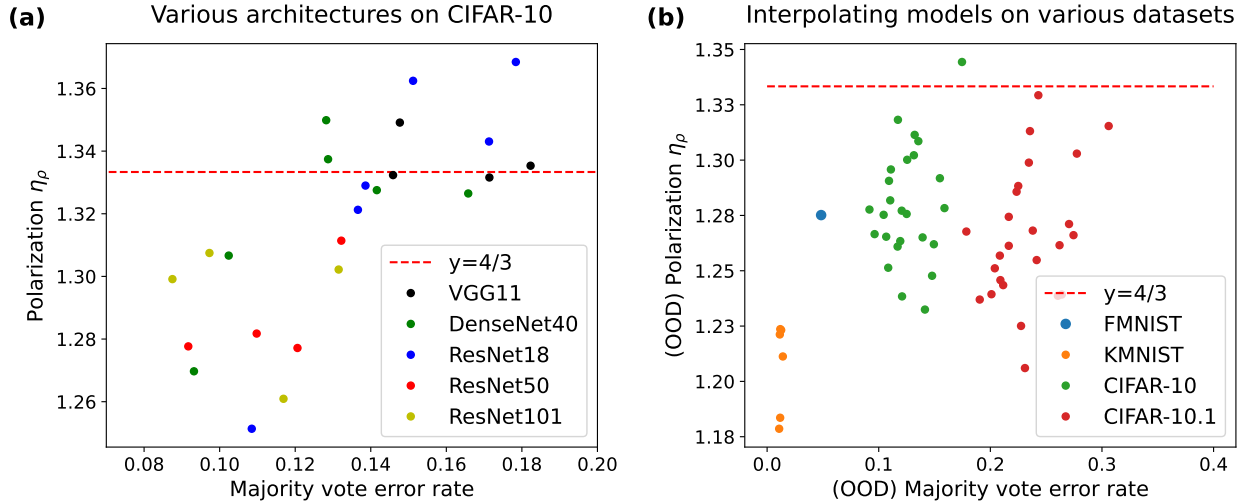


Figure 3.2: Polarization η_ρ obtained (a) from various architectures trained on CIFAR-10 and (b) only from interpolating classifiers trained on various datasets. Red dashed line indicates $y = 4/3$. In subplot (b), we observe that the polarization of all interpolating models except one are smaller than $4/3$, which aligns with Conjecture 3.1.

Remark. We emphasize that values for η_ρ that are larger than $\frac{4}{3}$ does *not* contradict Theorem 3.1. This happens when the non-constant second term in (3.9) is larger than $\frac{4}{3}$, which is often the case for classifiers which are not interpolating (or, indeed, that underfit or perform poorly).

Definition 3.2 (INTERPOLATING, [BHMM19]). *A classifier is **interpolating** if it achieves an accuracy of 100% on the training data.*

Putting Theorem 3.1 and the consistent empirical trend shown in Figure 3.2(b) together, we propose the following conjecture.

Conjecture 3.1 (NEURAL POLARIZATION LAW). *The polarization of ensembles comprised of independently trained interpolating neural networks is smaller than $\frac{4}{3}$.*

3.4 Entropy-restricted ensembles

In this section, we first present an upper bound on the majority vote error rate, $L(h_\rho^{\text{MV}})$, in Theorem 3.2, using our notion of polarization η_ρ which we introduced and defined in the previous section. Then, we present Theorems 3.3 and 3.4 which are the main elements in obtaining tighter upper bounds on $L(h_\rho^{\text{MV}})$. Figure 3.3 shows our proposed bound offers a significant improvement over state-of-the-art results. The new upper bounds are inspired from the fact that classifier prediction probabilities tend to concentrate on a small number of labels, rather than be uniformly spread over all the possible labels. This is analogous to the phenomenon of *neural collapse* [Kot22]. As an example, in the context of a computer vision model, when presented with a photo of a dog, one might expect that a large portion of reasonable models might classify the photo as an animal other than a dog, but not as a car or an airplane.

We start by stating an upper bound on the majority vote error, $L(h_\rho^{\text{MV}})$ as a function of polarization η_ρ . This upper bound is tighter (smaller) than the previous bound in inequality (3.6) when the polarization is lower than 2, which is the case for competent ensembles. The proof is deferred to Appendix B.2.2.

Theorem 3.2. *For an ensemble ρ of K -class classifiers,*

$$L(h_\rho^{\text{MV}}) \leq \frac{2\eta_\rho(K-1)}{K} \left(\mathbb{E}_\rho[L(h)] - \frac{1}{2}\mathbb{E}_{\rho^2}[D(h, h')] \right),$$

where η_ρ is the polarization of the ensemble ρ .

Based on the upper bound stated in Theorem 3.2, we add a restriction on the entropy of constituent classifiers to obtain Theorem 3.3. The theorem provides a tighter scalable bound that does *not* have explicit dependency on the total number of labels, with a small cost in terms of the entropy of constituent classifiers. The proof of Theorem 3.3 is deferred to Appendix B.2.3.

Theorem 3.3. *Let ρ be any η -polarized ensemble of K -class classifiers that satisfies $\mathbb{P}_\rho(h(x) \notin A(x)) \leq \Delta$, where $y \in A(x) \subset [K]$ and $|A(x)| \leq M$, for all data points $(x, y) \in \mathcal{D}$. Then, we have*

$$L(h_\rho^{\text{MV}}) \leq \frac{2\eta(M-1)}{M} \left[\left(1 + \frac{\Delta}{M-1}\right) \mathbb{E}_\rho[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right].$$

While Theorem 3.3 might provide a tighter bound than prior work, coming up with pairs (M, Δ) that satisfy the constraint is not an easy task. This is not an issue for a discrete ensemble, however. If ρ is a discrete distribution of N classifiers, then we observe that the assumption of Theorem 3.3 must always hold with $(M, \Delta) = (N+1, 0)$. We state this as the following corollary.

Corollary 3.1 (FINITE ENSEMBLE). *For an ensemble ρ that is a weighted sum of N classifiers, we have*

$$L(h_\rho^{\text{MV}}) \leq \frac{2\eta_\rho N}{N+1} \left(\mathbb{E}_\rho[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right), \quad (3.10)$$

where η_ρ is the polarization of the ensemble ρ .

See Figure 3.3, which provides empirical results that compare the bound in Corollary 3.1 with the C-bound in inequality (3.3), and with inequality (3.6) proposed in [TKY⁺24]. We can observe that the new bound in Corollary 3.1 is strictly tighter than the others. For more details on these empirical results, see Appendix B.3.

Although the bound in Corollary 3.1 is tighter than the bounds from previous studies, it's still not tight enough to use it as an estimator for $L(h_\rho^{\text{MV}})$. In the following theorem, we use a stronger condition on the entropy of an ensemble to obtain a tighter bound. The proof is deferred to Appendix B.2.4.

Theorem 3.4. *For any η -polarized ensemble ρ that satisfies*

$$\frac{1}{2} \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) \neq Y, h(X) \neq h'(X))] \leq \varepsilon \mathbb{E}_{\mathcal{D}} [\mathbb{P}_\rho(h(X) \neq Y)], \quad (3.11)$$

we have

$$L(h_\rho^{\text{MV}}) \leq \eta \left[(1 + \varepsilon) \mathbb{E}_\rho[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right].$$

The condition (3.11) can be rephrased as follows: compared to the error $\mathbb{P}_\rho(h(x) \neq y)$, the entropy of the distribution of wrong predictions is small, and it is concentrated on a small number of labels. A potential problem is that one must know or estimate the smallest possible value of ε in advance. At least, we can prove that $\varepsilon = \frac{K-2}{2(K-1)}$ always satisfies the condition (3.11) for an ensemble of K -class classifiers. The proof is deferred to Appendix B.2.4.

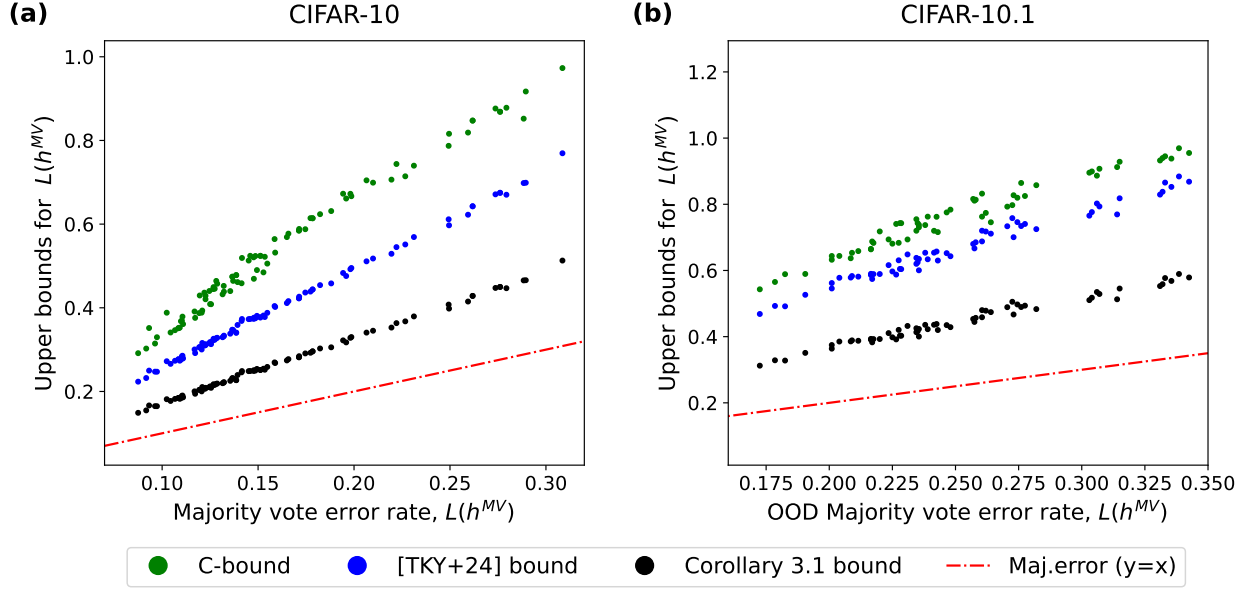


Figure 3.3: Comparing our new bound from Corollary 3.1 (colored black), which is the right hand side of inequality (3.10), with bounds from previous studies. Green corresponds to the C-bound in inequality (3.3), and blue corresponds to the right hand side of inequality (3.6). ResNet18, ResNet50, ResNet101 models with various sets of hyperparameters are trained on CIFAR-10 then tested on (a) the out-of-sample CIFAR-10, (b) an out-of-distribution dataset, CIFAR-10.1.

Corollary 3.2. *For any η -polarized ensemble ρ of K -class classifiers, we have*

$$L(h_\rho^{\text{MV}}) \leq \eta \left[\left(1 + \frac{K-2}{2(K-1)} \right) \mathbb{E}_\rho[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right].$$

Naturally, this ε is not good enough for our goal. We discuss more on how to estimate the smallest possible value of ε in the following section.

3.5 A universal law for ensembling

In this section, our goal is to predict the majority vote error rate of an ensemble with large number of classifiers by just using information we can obtain from an ensemble with a small number, e.g., three, of classifiers. Among the elements in the bound in Theorem 3.4,

$$\eta \left[(1 + \varepsilon) \mathbb{E}_\rho[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right],$$

we plug in $\eta = \frac{4}{3}$ as a result of Theorem 3.1; and since $\mathbb{E}_\rho[L(h)]$ is invariant to the number of classifiers, it remains to predict the behavior of $\mathbb{E}_{\rho^2}[D(h, h')]$ and the smallest possible

value of ε , $\varepsilon_\rho = \frac{\mathbb{E}_{\mathcal{D}}[\mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) \neq Y, h(X) \neq h'(X))]}{2\mathbb{E}_{\mathcal{D}}[\mathbb{P}_\rho(h(X) \neq Y)]}$. Since the denominator $\mathbb{E}_{\mathcal{D}}[\mathbb{P}_\rho(h(X) \neq Y)] = \mathbb{E}_\rho[L(h)]$ is invariant to the number of classifiers, and the numerator resembles the disagreement between classifiers, ε_ρ is expected to follow a similar pattern as $\mathbb{E}_{\rho^2}[D(h, h')]$. Note that the numerator of ε_ρ has the same form as the disagreement, differing by only one less label. Both are V -statistics that can be expressed as a multiple of a U -statistic, as shown in equation (B.10). In the next theorem, we show that the disagreement for a finite number of classifiers can be expressed as the sum of a hyperbolic curve and an unbiased random walk. Here, $[x]$ denotes the greatest integer less than or equal to x and $\mathcal{D}[0, 1]$ is the Skorokhod space on $[0, 1]$ (see Appendix B.2.6). The proof is deferred to Appendix B.2.5.

Theorem 3.5. *Let ρ_N denote an empirical distribution of N independent classifiers $\{h_i\}_{i=1}^N$ sampled from a distribution ρ and $\sigma_1^2 = \text{Var}_{h \sim \rho}(\mathbb{E}_{h' \sim \rho} \mathbb{P}_{\mathcal{D}}(h(X) \neq h'(X)))$. Then, there exists $D_\infty > 0$ such that*

$$\mathbb{E}_{(h, h') \sim \rho_N^2}[D(h, h')] = \left(1 - \frac{1}{N}\right) \left(D_\infty + \frac{2}{\sqrt{N}} Z_N\right),$$

where $\mathbb{E}Z_N = 0$, $\text{Var}Z_N \rightarrow \sigma_1^2$ and $\{\frac{\sqrt{t}}{\sigma_1} Z_{[Nt]}\}_{t \in [0, 1]}$ converges weakly to a standard Wiener process in $\mathcal{D}[0, 1]$ as $N \rightarrow \infty$.

Theorem 3.5 suggests that the disagreement within N classifiers, $\mathbb{E}_{\rho_N^2}[D(h, h')]$, can be approximated as $\frac{N-1}{N} D_\infty$. From the disagreement within $M (\ll N)$ classifiers, D_∞ can be approximated as $\frac{M}{M-1} \mathbb{E}_{\rho_M^2}[D(h, h')]$, and therefore we get

$$\mathbb{E}_{\rho_N^2}[D(h, h')] \approx \frac{N-1}{N} \cdot \frac{M}{M-1} \mathbb{E}_{\rho_M^2}[D(h, h')]. \quad (3.12)$$

Assume that we have three classifiers sampled from ρ . We denote the average error rate, the disagreement, and the ε_ρ from these three classifiers by $\mathbb{E}_3[L(h)]$, $\mathbb{E}_3[D(h, h')]$, and ε_3 , respectively. Then, from Theorem 3.4 and approximation (3.12) (which applies to both disagreement and ε_ρ), we estimate the majority vote error rate of N classifiers from ρ as the following:

$$\begin{aligned} L(h_\rho^{\text{MV}}) &\lesssim \frac{4}{3} \left[\left(1 + \frac{N-1}{N} \cdot \frac{3}{2} \cdot \varepsilon_3\right) \mathbb{E}_3[L(h)] - \frac{N-1}{N} \cdot \frac{3}{2} \cdot \frac{1}{2} \mathbb{E}_3[D(h, h')] \right] \\ &= \frac{4}{3} \left[\mathbb{E}_3[L(h)] + \frac{3(N-1)}{2N} \left(\varepsilon_3 \mathbb{E}_3[L(h)] - \frac{1}{2} \mathbb{E}_3[D(h, h')] \right) \right]. \end{aligned} \quad (3.13)$$

Alternatively, we can use the polarization measured from three classifiers, η_3 , instead of $\eta = \frac{4}{3}$, to obtain:

$$L(h_\rho^{\text{MV}}) = \eta_3 \left[\mathbb{E}_3[L(h)] + \frac{3(N-1)}{2N} \left(\varepsilon_3 \mathbb{E}_3[L(h)] - \frac{1}{2} \mathbb{E}_3[D(h, h')] \right) \right]. \quad (3.14)$$

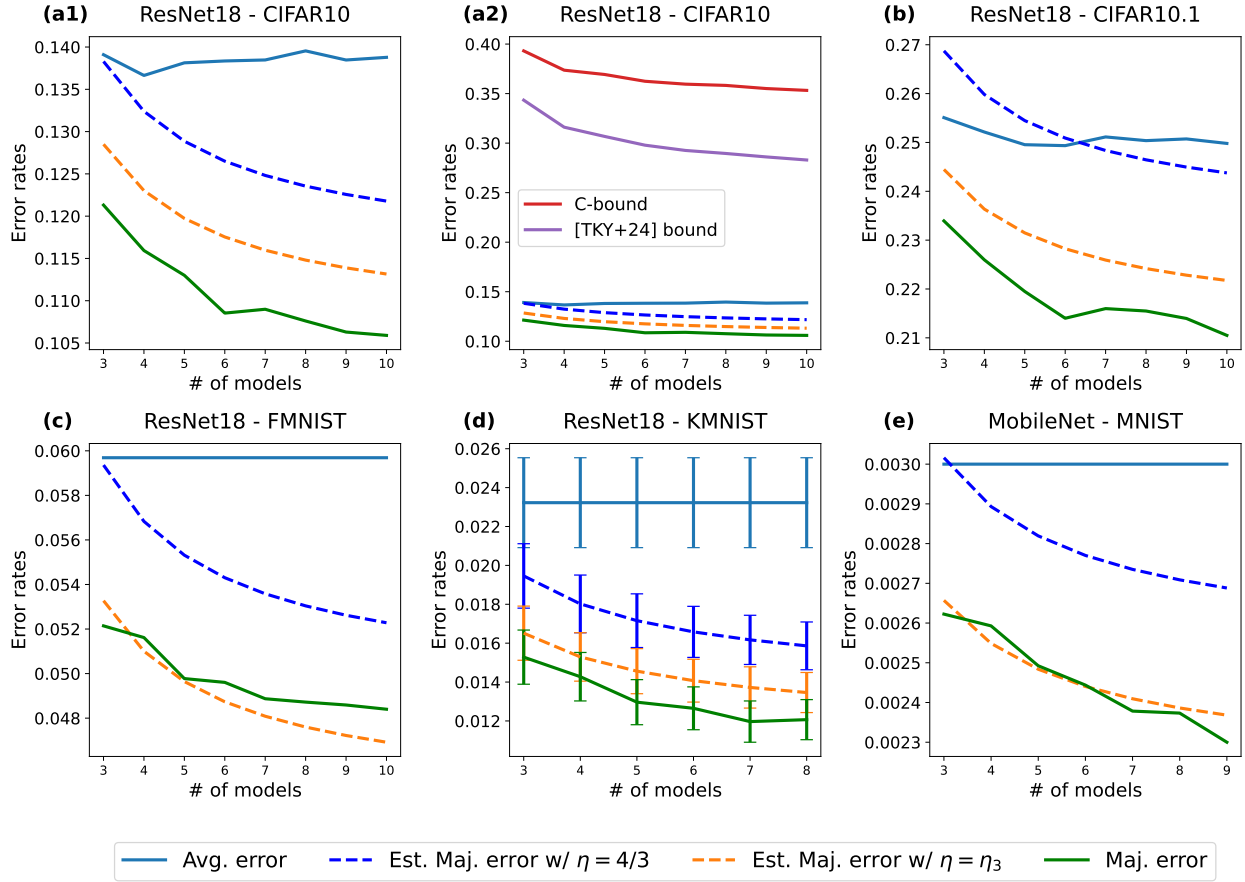


Figure 3.4: Comparing the estimated (extrapolated) majority vote error rates in equation (3.13) (blue-dashed lines) and (3.14) (orange-dashed lines) with the true majority vote error rate (green solid line) for each number of classifiers. The solid sky-blue line corresponds to the average error rate of constituent classifiers. Subplots (a1), (b), (c), (d), (e) show the results from different pairs of (classification model, dataset). Subplot (a2) overlays the right hand side of inequality (3.3) (C-bound, colored red) and inequality (3.6) ([TKY+24] bound, colored purple) on the subplot (a1). These two quantities from previous studies are much larger compared to the average error rate. We see the same pattern for other (architecture, dataset) pairs, which we therefore omit from the plot. For more details on these empirical results, see Appendix B.3.

Figure 3.4 presents empirical results that compare the estimated (extrapolated) majority vote error rates in equations (3.13) and (3.14) with the true majority vote error for each number of classifiers. ResNet18 models are tested on four different dataset: CIFAR-10, CIFAR-10.1, Fashion-MNIST [XRV17] and Kuzushiji-MNIST [CBIK⁺18] where the models are trained on the corresponding train data. MobileNet [How17] is trained and tested on the MNIST [Den12] dataset. Not only do the estimators show significant improvement compared to the bounds introduced in Section 3.2.2, we observe that the estimators are very close to the actual majority vote error rate; and thus the estimators have practical usages, unlike the bounds from previous studies. In Figure 3.4(a2), existing bounds (3.3) and (3.6) are much larger compared to the average error rate. This is also the case for (architecture, dataset) pairs of other subplots.

3.6 Discussion and conclusion

This work addresses the question: how does the majority vote error rate change according to the number of classifiers? While this is an age-old question, it is one that has received renewed interest in recent years. On the journey to answering the question, we introduce several new ideas of independent interest. (1) We introduced the polarization η_ρ , of an ensemble of classifiers. This notion plays an important role throughout this chapter and appears in every upper bound presented. Although Theorem 3.1 gives some insight into polarization, our conjectured neural polarization law (Conjecture 3.1) is yet to be proved or disproved, and it provides an exciting avenue for future work. (2) We proposed two classes of ensembles whose entropy is restricted in different ways. Without these constraints, there will always be examples that saturate even the least useful majority vote error bounds. We believe that accurately describing how models behave in terms of the entropy of their output is key to precisely characterizing the behavior of majority vote, and likely other ensembling methods.

Throughout this chapter, we have theoretically and empirically demonstrated that polarization is fairly invariant to the hyperparameters and architecture of classifiers. We also proved a tight bound for majority vote error, under an assumption with another quantity ε , and we presented how the components of this tight bound behave according to the number of classifiers. Altogether, we have sharpened bounds on the majority vote error to the extent that we are able to identify the trend of majority vote error rate in terms of number of classifiers.

We close with one final remark regarding the metrics used to evaluate an ensemble. Majority vote error rate is the most common and popular metric used to measure the performance of an ensemble. However, it seems unlikely that a practitioner would consider an ensemble to have performed adequately if the majority vote conclusion was correct, but was only reached by a relatively small fraction of the classifiers. With the advent of large language models, it is worth considering whether the majority vote error rate is still as valuable. The natural alternative in this regard is the probability $\mathbb{P}_\rho(W_\rho > 1/2)$, that is, the probability that at least

half of the classifiers agree on the correct answer. This quantity is especially well-behaved, and it frequently appears in our proofs. (Indeed, every bound presented in this work serves as an upper bound for $\mathbb{P}_\rho(W_\rho > 1/2)$.) We conjecture that this quantity is useful much more generally.

Chapter 4

Loss-Robust Quantile Prediction for Proper Losses

4.1 Introduction

Probabilistic forecasting, i.e., forecasting the distribution of a target variable, has become a popular framework across a variety of prediction tasks [GK14, VBD⁺21, SWM⁺18, CRL⁺22]. In contrast to point predictions, probabilistic forecasts provide a richer description of uncertainty and offer end-users greater information with which to evaluate risk and guide effective decision-making [Raf16]. This flexibility is particularly valuable in settings in which a single prediction is shown to a diverse set of users with varying and unknown objectives, as it allows users to extract the information most relevant to their needs.

A conventional approach for training and evaluating a probabilistic forecast is to measure its distance from the true distribution using a proper loss [GRWG05, GR07]. In general, we say that a loss is proper if (in expectation) it is minimized by predicting the true distribution of the outcome. There are a number of popular proper losses in the literature, and a key difficulty is determining which loss one should use for training. Indeed, while all proper losses share the same global optimum, they penalize distance from the ground truth asymmetrically, and thus obtaining a near minimizer in one loss does not guarantee good performance on other metrics [BRT25].

In response to this, a recent line of work aims to construct omnipredictors that achieve optimal performance on multiple losses simultaneously [GKR⁺22, GKR23, GJRR24, GOR⁺24, NRRX25, LRS25, OKK25, GT25]. Much of this work has focused on binary prediction tasks. There, it has been shown that given a baseline class of prediction functions one can construct a single unified predictor that approximately matches the performance of the best baseline on every proper loss simultaneously [GJRR24, OKK25, GT25]. Notably, the error rates achieved in these works are identical (up to logarithmic factors) to the standard error rates for minimizing a single loss function ($\tilde{O}(\sqrt{1/T})$) in the online setting

over T timesteps, and $\tilde{O}(\sqrt{1/n})$ in the offline setting over n samples). Thus, omniprediction offers the potential to provide a practical path to improving existing learning algorithms. On the other hand, results for multiclass and continuous targets are more modest. Prior work in [GKR⁺22, GOR⁺24, LRS25] gives algorithms for multiclass or continuous outcome omniprediction over restricted classes of losses satisfying various convexity and Lipschitz conditions, while [NRRX25] considers multiclass Bregman losses that are separable in the predicted probability for each class. In all cases, omniprediction is achieved at significantly slower rates than in the binary setting. For more general losses, [HTY26] propose a method for obtaining omniprediction over all proper losses in the multiclass setting. However, its error rate deteriorates rapidly as the number of classes grows, with realized rates of $(k/n)^{1/(k+1)}$ and $(k/T)^{1/(k+1)}$ for online/offline k -class classification, respectively.

To achieve more practical methods, we pursue omniprediction guarantees for simplified probabilistic forecasts expressed as dense grids of quantile predictions for the outcome. Our motivation for considering such quantile forecasts is two-fold. First, on a technical level, proper losses for quantiles admit a decomposition as mixtures over a smaller two-parameter class of elementary losses. This will allow us to focus our algorithms on a small, tractable set of losses and thus enable the development of efficient omniprediction algorithms that mirror those for the binary prediction setting. Second, more practically, quantile forecasts are a popular way to summarize uncertainty in many real-world forecasting domains [HF16, CRL⁺22]. As a demonstration, in Section 4.4, we will show how our methods can be used to improve quantile-level forecasts of COVID-19 hospitalizations made by expert forecasting groups during the pandemic.

The remainder of this article is structured as follows. Section 4.2 formally defines our prediction framework and presents the key structural results for proper quantile losses used throughout this article. Section 4.3 develops our main method, which is an online forecasting algorithm for constructing quantile predictions that achieve low regret on all (sufficiently smooth) proper losses simultaneously. Analogous to results for the binary setting, we show that the omniprediction error of this method over T timesteps scales as $\tilde{O}(\sqrt{1/T})$ and thus matches rates for learning under a single loss (up to logarithmic factors). Finally, Section 4.4 gives our main empirical results where we show that this procedure can be used to improve quantile forecasts made during the COVID-19 pandemic and we conclude in Section 4.5 with a discussion of limitations and directions for future work. Additional technical details are deferred to the appendix, including proofs of all theoretical results, efficient subroutines for faster implementation of our algorithm, and additional experimental results and plots.

4.2 Background and preliminaries

4.2.1 Problem statement

We consider an online adversarial setting in which at each timestep $t \in [T]$ the goal is to forecast an outcome Y_t whose distribution is allowed to depend on the entire history of

previous outcomes and predictions. For simplicity, we assume that Y_t lies in $[0, 1]$ though our results readily extend to any bounded interval by translation and rescaling.

Given quantile levels $0 < \tau_1 < \tau_2 < \dots < \tau_N < 1$, our forecasts take the form of a vector $q(t) = (q_1(t), \dots, q_N(t))$ where $q_n(t)$ denotes our estimate of the τ_n quantile of Y_t . We evaluate the performance of these predictions by comparing them to a finite set of B base forecasters $\mathcal{F} = \{\{f_b^n(t)\}_{n \in [N], t \in [T]} : b \in [B]\}$, where $f_b^n(t) \in [0, 1]$ denotes the forecast of the τ_n -quantile made by base forecaster f_b at timestep t .

As motivating examples, one may consider problems in which $\mathcal{F}$ is a collection of parametric functions mapping some covariates X_t to Y_t or problems in which $\mathcal{F}$ contains candidate forecasts from many different baseline learning algorithms. In either case, at every timestep t we will observe the previous sequence of outcomes $(Y_s)_{s < t}$ and the current and preceding base forecasts $\{\{f_b^n(s)\}_{n \in [N], s \leq t} : b \in [B]\}$ and our goal will be to construct $q(t)$ so that it matches the performance of the best base forecaster on every loss ℓ in some collection $\mathcal{L}$. More precisely, following [GKR⁺22], we aim to minimize the omniprediction error defined as

$$\sup_{\ell \in \mathcal{L}, b \in [B]} \frac{1}{T} \sum_{t=1}^T \ell(q(t), Y_t) - \ell(f_b(t), Y_t), \quad (4.1)$$

where $\mathcal{F} = \{f_b\}_{b=1}^B$ and $f_b(t) := (f_b^1(t), \dots, f_b^N(t))$. It is important to emphasize that in this definition the comparator forecaster in $\mathcal{F}$ is allowed to depend on the loss $\ell \in \mathcal{L}$. As a result, to achieve low omniprediction error it is not sufficient to set $q(t)$ to be a fixed element of $\mathcal{F}$ as the best base forecaster may vary between losses.

In this article we will be interested in pursuing low omniprediction error with respect to the class of proper losses. Broadly, we say that a loss function is proper for forecasting a given set of quantiles if it is minimized by forecasting the true quantiles of the target.¹ This is formalized in the following definition.

Definition 4.1 (Proper loss for quantile levels). *For a distribution P_Y on $[0, 1]$, denote its τ -level quantile by $q_\tau^*(P_Y) := \inf\{q \in [0, 1] : \mathbb{P}_{Y \sim P_Y}(Y \geq q) \geq \tau\}$. A loss function $\ell : [0, 1]^N \times [0, 1] \rightarrow \mathbb{R}_{\geq 0}$ is referred to as a **proper loss for quantile levels** $0 < \tau_1 < \tau_2 < \dots < \tau_N < 1$ if*

$$(q_{\tau_n}^*(P_Y))_{n=1}^N \in \arg \min_{q \in \mathbb{R}^N} \mathbb{E}_{Y \sim P_Y}[\ell(q, Y)], \quad \text{for all distributions } P_Y \text{ supported on } [0, 1].$$

By definition, the class of proper losses is the largest class of losses that share the true quantiles as their common minimizer and thus is the largest class over which we can reasonably expect to achieve low omniprediction error.

¹Losses with this property are also sometimes referred to as consistent losses or consistent scoring functions.

4.2.2 Existing characterizations of proper losses

In order to make our omniprediction objective tractable, we will rely on structural characterizations of proper losses for quantiles. Such characterizations have a long history in the forecasting literature dating back to the pioneering work of [Sav71]. We will in particular rely on results for quantiles developed in [Tho79, Sae00, Gne11, EGJK16] which show that any (sufficiently smooth) proper loss for a single quantile level can be decomposed as a mixture over a one-dimensional family of elementary losses. To outline this result, [EGJK16] summarizes the characterization of proper loss for single quantile level from previous works as follows:

Characterization. For any non-decreasing function $g : [0, 1] \rightarrow \mathbb{R}$, the loss function $\ell : [0, 1] \times [0, 1] \rightarrow \mathbb{R}_{\geq 0}$ given by

$$\ell(q, y) = (\mathbf{1}(y < q) - \tau)(g(q) - g(y)) \quad (4.2)$$

is proper for the τ -level quantile. Moreover, this representation is **exhaustive** in the sense that any proper loss function for the τ -level quantile that satisfies the mild regularity conditions (L0) $\ell(q, y) \geq 0$ with equality if $q = y$, (L1) ℓ is continuous, and (L2) ℓ is twice continuously differentiable for $q \neq y$ can be expressed in this form.

Let $\mathcal{L}_\tau$ denote the class of loss functions of the form (4.2) with g being a left-continuous, non-decreasing function. Then, [EGJK16] shows that every loss in $\mathcal{L}_\tau$ can be decomposed as a mixture of so-called elementary losses as follows.

Theorem 4.1 (Theorem 1a in [EGJK16]). *Fix any $\tau \in (0, 1)$. For any loss function $\ell \in \mathcal{L}_\tau$, there exists a non-negative Lebesgue–Stieltjes measure H on $[0, 1]$, given by $dH(\theta) = dg(\theta)$, such that for all $q, y \in [0, 1]$,*

$$\ell(q, y) = \int_0^1 \ell_{\tau, \theta}(q, y) dH(\theta), \quad (4.3)$$

where for any $\theta \in [0, 1]$, $\ell_{\tau, \theta}$ is the elementary loss

$$\ell_{\tau, \theta}(q, y) = (\mathbf{1}(y < q) - \tau)(\mathbf{1}(\theta < q) - \mathbf{1}(\theta < y)) = \begin{cases} 1 - \tau, & y \leq \theta < q, \\ \tau, & q \leq \theta < y, \\ 0, & \text{otherwise.} \end{cases} \quad (4.4)$$

It may be instructive to note that for any $\tau, \theta \in [0, 1]$ the elementary loss $\ell_{\tau, \theta}$ is itself proper. To see this, fix any $\tau, \theta \in (0, 1)$ ² and note that by definition $\ell_{\tau, \theta}$ essentially only admits two forecasts. Namely, we may either forecast a value $q > \theta$ or a value $q \leq \theta$. Which of these forecasts is optimal will depend on the sign of

$$\mathbb{E}_{Y \sim P_Y}[\ell_{\tau, \theta}(1, Y)] - \mathbb{E}_{Y \sim P_Y}[\ell_{\tau, \theta}(0, Y)] = (1 - \tau)\mathbb{P}(y \leq \theta) - \tau\mathbb{P}(y > \theta),$$

²The cases $\tau \in \{0, 1\}$ and $\theta \in \{0, 1\}$ can be proven similarly and are omitted.

which is nonnegative (resp. nonpositive) exactly when θ is below (resp. above) the τ -quantile of P_Y .

Decision-making interpretation of elementary losses.

Following the exposition in [MIT⁺25], one may also interpret $\ell_{\tau,\theta}$ in terms of a decision-making task in which one must decide whether to take a preventative action against an adverse event. More formally, let C denote the cost of taking this action (e.g., expanding hospital capacity) and L denote the price paid if the adverse event occurs (e.g., COVID-19 cases surge) and no preventative action was taken. Let $a = \mathbb{1}\{\text{action taken}\}$ and $o = \mathbb{1}\{\text{adverse event occurs}\}$ denote binary indicators for the action and outcome, respectively, and define a utility s expressing the payoffs for these outcomes as in Table 4.1. Finally, suppose the adverse event occurs exactly when some outcome Y (e.g., COVID-19 cases) exceeds a given threshold θ (e.g., the hospital's current capacity). Then, if we take the preventative action if and only if the forecast quantile q exceeds θ , one can check that

Table 4.1: Payoff table

$s(a, o)$	$o = 1$	$o = 0$
$a = 1$	C	C
$a = 0$	L	0

$$s(\mathbb{1}\{q > \theta\}, \mathbb{1}\{Y > \theta\}) - s(\mathbb{1}\{Y > \theta\}, \mathbb{1}\{Y > \theta\}) = L \cdot \ell_{1-C/L, \theta}(q, Y),$$

and thus up to renormalization correctly forecasting the $(1 - C/L)$ -level quantile of Y is equivalent to choosing the correct action.

To extend beyond single quantile forecasts we will utilize a result of [FZ16] which shows that proper losses for vectors of quantiles decompose as sums of proper losses for single quantiles.

Theorem 4.2 (Simplified version of Proposition 4.2 in [FZ16]). *Fix any $0 < \tau_1 < \dots < \tau_N < 1$ and let $\ell : [0, 1]^N \times [0, 1] \rightarrow \mathbb{R}_{\geq 0}$ be a proper loss for quantile levels $\{\tau_n\}_{n=1}^N$ that satisfies mild regularity conditions (See Appendix C.6.1). Then, there exists losses $\ell_1, \dots, \ell_N : [0, 1] \times [0, 1] \rightarrow \mathbb{R}_{\geq 0}$ such that ℓ_n is a proper loss for the τ_n -level quantile and*

$$\ell(q_1, \dots, q_N, y) = \sum_{n=1}^N \ell_n(q_n, y). \quad (4.5)$$

Now, define the following two classes of loss functions $\ell : [0, 1]^N \times [0, 1] \rightarrow \mathbb{R}_{\geq 0}$:

$$\mathcal{L} := \left\{ \ell = \sum_{n=1}^N \ell_n \mid \ell_n(q_n, y) = (\mathbb{1}(y < q_n) - \tau_n)(g_n(q_n) - g_n(y)) \right\}, \quad \mathcal{L}_1 := \{\ell \in \mathcal{L} \mid \|\ell\|_\infty \leq 1\} \quad (4.6)$$

where each g_n is non-decreasing and left-continuous. Putting the results above together, any loss function $\ell \in \mathcal{L}$ can be expressed as $\ell(q_1, \dots, q_N, y) = \sum_{n=1}^N \int_{-\infty}^{\infty} \ell_{\tau_n, \theta}(q_n, y) dH_n(\theta)$ for some non-negative mixing measures H_n and is a proper loss for quantile levels $\{\tau_n\}_{n=1}^N$.

Moreover, this representation is **exhaustive**: $\mathcal{L}$ contains **all** proper losses for quantile levels $\{\tau_n\}_{n=1}^N$ that satisfies mild regularity conditions. Finally, we add the additional restriction in $\mathcal{L}_1$ that ℓ is bounded because this will give a common scale to our definition of omniprediction and ensure that the mixing distributions have bounded total measure.

4.3 Quantile omniprediction for proper losses

4.3.1 Discretization

We now turn to our main theoretical results which develop an algorithm for constructing quantile forecasts that achieves low omniprediction error with respect to the loss class $\mathcal{L}_1$. Our first step towards deriving this method will be to reduce from optimization over all losses in $\mathcal{L}_1$ to optimization over a finite class. To accomplish this, we will combine the characterizations of proper losses from the previous section with a discretization lemma that reduces the number of elementary losses that we need to consider to a finite set. For simplicity, in what follows we will assume that the outcomes Y_t and base forecasts $f_b^n(t)$ lie in the discrete grid $\{0, \frac{1}{m}, \dots, 1\}$, for some large fixed value $m \in \mathbb{N}$. This assumption is relatively minor and is made primarily for technical convenience. Indeed, our results can be readily extended to the case where Y_t is continuously distributed with a bounded density by incurring an extra additive error of size $O(1/m)$. Now, let $\theta_i = \frac{i}{m} - \frac{1}{2m}$ for $i \in [m]$. The following lemma shows that to control the omniprediction error over the class $\mathcal{L}_1$ it suffices to control the error over the finite collection of elementary losses $\{\ell_{\tau_n, \theta_i}\}_{i \in [m], n \in [N]}$. This discretization scheme is visualized in Figure 4.1 and a proof of this result can be found in Appendix C.1.1

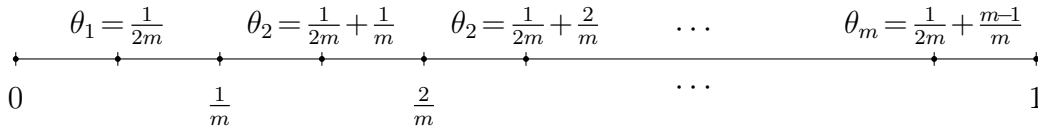


Figure 4.1: Visualization of the discretization on the real line. Outcomes Y_t and forecasts $f_b^n(t)$ take values in the grid $\{0, 1/m, \dots, 1\}$ that is interleaved by the values of θ_i .

Lemma 4.1. *Let $\mathcal{L}_1$ denote the class of bounded proper losses defined in (4.6) and suppose $Y_t, q_n(t), f_b^n(t) \in \{0, 1/m, \dots, 1\}$ for all $t \in [T]$, $n \in [N]$, and $b \in [B]$. Then,*

$$1 \leq \frac{\sup_{\ell \in \mathcal{L}_1, b} \frac{1}{T} \sum_{t=1}^T [\ell(q(t), Y_t) - \ell(f_b(t), Y_t)]}{\sup_{i, n, b} \frac{1}{T} \sum_{t=1}^T [\ell_{\tau_n, \theta_i}(q_n(t), Y_t) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)]} \leq 2.$$

4.3.2 Algorithm

We now introduce our main method, Algorithm 1. Due to Lemma 4.1, it suffices to minimize the following objective:

$$\sup_{i \in [m], n \in [N], b \in [B]} \frac{1}{T} \sum_{t=1}^T [\ell_{\tau_n, \theta_i}(q_n(t), Y_t) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)]. \quad (4.7)$$

To accomplish this, we will draw on an algorithmic framework for multiobjective learning developed in [LNPR22] and popularized in a number of recent works on omniprediction and related problems [NRRX25, GJRR24, LRS25, GT25]. The core idea is to consider a two-player game in which one player constructs a mixture of the elementary loss functions ℓ_{τ_n, θ_i} and the other player best responds with a (randomized) predictor that minimizes the worst-case loss gap on the mixture. Concretely, in Algorithm 1, this is implemented by having the first player use the well-known Hedge algorithm [Vov90, LW94, FS97] to learn weights $w_{ib}^n(t)$ over the quantiles levels τ_n , parameters θ_i , and baselines f_b^n appearing in (4.7) so that the time-average weighted mixture loss,

$$\frac{1}{T} \sum_{t=1}^T \sum_{i,n,b} w_{ib}^n(t) [\ell_{\tau_n, \theta_i}(q_n(t), Y_t) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)],$$

is guaranteed to be close to the supremum in (4.7). Then, letting Δ_m denote the set of distributions over $[m]$, the second player constructs their (randomized) forecast as $q(t) \sim \hat{P}_t$ where $\hat{P}_t$ solves the minmax program,

$$\hat{P}(t) = \arg \min_{P_1, \dots, P_N \in \Delta_m} \max_{P_y \in \Delta_m} \sum_{i=1}^m \sum_{n=1}^N \sum_{b=1}^B w_{ib}^n \mathbb{E}_{q_n \sim P_n, Y \sim P_y} [\ell_{\tau_n, \theta_i}(q_n, Y) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y)]. \quad (4.8)$$

By definition, this forecast minimizes the mixture loss over the worst-case (and potentially adversarially chosen) value of Y_t .

As anticipated, the following theorem shows that Algorithm 1 achieves our desired omniprediction guarantee. Proof of this result can be found in Appendix C.1.3.

Theorem 4.3. *With learning rate $\eta = c\sqrt{(\log mNB)/T}$, the predictions $\hat{P}(t) = (\hat{P}_n(t))_{n=1}^N$ from Algorithm 1 achieve the omniprediction error bound*

$$\sup_{\ell \in \mathcal{L}_1, b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q_n \sim \hat{P}_n(t)} [\ell((q_n)_{n=1}^N, Y_t) - \ell(f_b(t), Y_t)] \leq 2 \left(c + \frac{1}{c} \right) \left(\sqrt{\frac{\log mNB}{T}} \right).$$

A critical issue when implementing Algorithm 1 is solving the minmax program (4.8) in an efficient manner. Our detailed procedure for accomplishing this is given in Appendix C.4. As a summary of those results, the following theorem states the key structural properties of our

Algorithm 1 Omniprediction under all proper losses for multiple quantile levels

- 1: **Data:** Data stream $(Y_t)_{t=1}^T$, quantile levels $\{\tau_n\}_{n=1}^N$, grid size $m \in \mathbb{N}$, learning rate $\eta > 0$, base forecasters $\mathcal{F} = \{f_b\}_{b=1}^B$.
- 2: **Initialize:** $w_{ib}^n(1) = \frac{1}{mnB}$, for all $i \in [m], n \in [N], b \in [B]$;
- 3: **for** $t = 1, \dots, T$ **do**
- 4: **1) Make prediction at time t :**

$$\hat{P}(t) = \arg \min_{P_1, \dots, P_N \in \Delta_m} \max_{P_y \in \Delta_m} \sum_{i=1}^m \sum_{n=1}^N \sum_{b=1}^B w_{ib}^n \mathbb{E}_{q_n \sim P_n, Y \sim P_y} [\ell_{\tau_n, \theta_i}(q_n, Y) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y)]; \quad (4.9)$$

- 5: **2) Update the weights:** For all i, n, b ,

$$w_{ib}^n(t+1) = \frac{w_{ib}^n(t) \exp(\eta(\mathbb{E}_{q_n \sim \hat{P}_n(t)}[\ell_{\tau_n, \theta_i}(q_n, Y_t)] - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)))}{\sum_{i', n', b'} w_{i'b'}^{n'}(t) \exp(\eta(\mathbb{E}_{q_{n'} \sim \hat{P}_{n'}(t)}[\ell_{\tau_{n'}, \theta_{i'}}(q_{n'}, Y_t)] - \ell_{\tau_{n'}, \theta_{i'}}(f_{b'}^{n'}(t), Y_t)))}.$$

- 6: **end for**
-

computed solution. Briefly, the first property states that although $\hat{P}(t)$ is randomized, this randomization is supported on only two nearby values. The second property states that the quantile forecasts produced by $\hat{P}(t)$ follow the anticipated order $q_1(t) \leq \dots \leq q_N(t)$, and the third property gives a bound on the range of $\hat{P}(t)$'s support. We provide more details and the proof of the theorem in Appendix C.1.2.

Theorem 4.4. *There exists a solution for $\hat{P}(t) = (\hat{P}_n(t))_{n=1}^N$ in Algorithm 1 such that:*

1. *For all $n \in [N]$, $\hat{P}_n(t)$ is supported on a maximum of two values.*
2. *There exists a coupling of $(\hat{P}_1(t), \dots, \hat{P}_N(t))$ such that any sampled value $q(t) \sim \hat{P}(t)$ is ordered as $q_1(t) \leq \dots \leq q_N(t)$.*
3. *For all $n \in [N]$, $\hat{P}_n(t)$ is supported on $[\min_{n \in [N], b \in [B]} f_b^n(t), \max_{n \in [N], b \in [B]} f_b^n(t)]$.*

The preceding theorem demonstrates that $\hat{P}(t)$ produces sensible quantile forecasts with only minimal randomization. Our second key result shows that $\hat{P}(t)$ can be computed in a highly efficient manner. This is crucial because although the minimax program in (4.8) is convex-concave, it involves optimization over a high-dimensional parameter space consisting of N distributions on $[m]$, making standard solution methods computationally intractable. As above, our detailed method for computing $\hat{P}(t)$ is deferred to Appendix C.4.

Proposition 4.1. *At each step of Algorithm 1, $\hat{P}(t)$ can be computed in $O(mNB)$ time. Consequently, Algorithm 1 can be implemented in $O(mNB)$ time complexity per step.*

4.3.3 Comparison with Separate Optimization of Quantile Levels

In this subsection, we illustrate an advantage of optimizing multiple quantile levels jointly as in Algorithm 1. In particular, in light of Theorem 4.2, one might ask whether running a separate omniprediction algorithm, analogous to Algorithm 1, for each quantile level individually would suffice. Appendix C.2 discusses single quantile omniprediction for proper losses in detail. In that case, the minimax program (4.9) would be replaced by

$$\hat{P}_n(t) = \arg \min_{P \in \Delta_m} \max_{P_y \in \Delta_m} \sum_{i=1}^m \sum_{b=1}^B w_{ib}^n \mathbb{E}_{q_n \sim P, Y \sim P_y} [\ell_{\tau_n, \theta_i}(q_n, Y) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y)], \quad \forall n \in [N].$$

By Theorem 4.4, Algorithm 1 outputs a vector of quantile estimates that is ordered according to the quantile levels. In contrast, no such guarantee holds when each quantile level is optimized separately. Moreover, simple procedures such as post-sorting the resulting quantile estimates may increase the loss and thus fail to provide an omniprediction guarantee (see Appendix C.6.2).

4.4 Evaluating the theory

To evaluate Algorithm 1, we consider a real epidemiological forecasting task from the COVID-19 forecasting hub, a repository of real-time forecasts made during the COVID-19 pandemic [CHW⁺22]. In particular, we consider forecasting weekly aggregated COVID-19 hospitalizations across 51 state-level series (50 states + DC) and the U.S. aggregate (US-agg), at $N = 23$ quantile levels and horizons of $h \in \{1, 2, 3, 4\}$ weeks ahead. We evaluate performance across all location-horizon pairs (each defining a time series). During the pandemic research groups throughout the country submitted forecasts to this repository. We utilize the predictions from $B = 18$ of these groups as the base forecasts for our algorithm. We compare our method against two ensembles constructed from the submitted forecasts by researchers during the pandemic, COVIDhub-Ens and JHUAPL-SLPEns, as well as Algorithm 3 (Hedge-QL, Appendix C.3). Hedge-QL applies the Hedge algorithm independently at each quantile level, forming predictions as weighted averages of base forecasts with weights chosen to minimize the pinball loss. This method acts as a simple baseline for ensembling quantile forecasts against which we compare our omniprediction procedure. We use 127 weeks of data from December 2020 to June 2023. Additional details about the data can be found in Appendix C.5.1.

Metrics We evaluate performance using three metrics. The first is the approximate cumulative omniprediction error at time t (referred to as the omni-error at time t):

$$\sup_{i,n,b} \frac{1}{t} \sum_{s=1}^t [\ell_{\tau_n, \theta_i}(q_n(s), Y_s) - \ell_{\tau_n, \theta_i}(f_b^n(s), Y_s)]. \quad (4.10)$$

Note that Lemma 4.1 shows that this quantity approximates the true omniprediction error (Eq. (4.1)) within a factor of 2. The second is the cumulative quantile loss at time t , defined as

$$\frac{1}{Y_{\max}} \frac{1}{t} \sum_{s=1}^t \sum_{n=1}^N \mathbb{E}_{q_n \sim \hat{P}_n(s)} [\rho_{\tau_n}(Y_s - q_n)],$$

where $\rho_{\tau}(x) = \tau x - \min\{x, 0\}$ is the canonical pinball loss for the τ -quantile and $Y_{\max}$ is the maximum value of the ground truth in the given time-series. To clarify terminology, we refer to the omni-error and cumulative quantile loss at the final time step T as the *total omni-error* and *total quantile loss*, respectively. Finally, as our last metric, we also consider the instantaneous normalized quantile loss at time t : $\frac{1}{Y_t} \sum_{n=1}^N \mathbb{E}_{q_n \sim \hat{P}_n(t)} [\rho_{\tau_n}(Y_t - q_n)]$.

Discretization As introduced in Section 4.3.1, we discretize both the predictions and the ground truth into a finite set of values. We use grid spacings from $\{2, 5, 10, 20, 50, 100, 500\}$ depending on the range of forecast values. This results in more than 500 grid points for all $208 = 52 \times 4$ time-series, and more than 1,000 grid points for over 150 time-series. Additional details on the choice of discretization are provided in Appendix C.5.1.

Learning rates Our algorithm and Hedge-QL each use a learning rate hyperparameter. Rather than tuning it explicitly, we evaluate each method over a range of learning rates and report results for the best-performing value on each metric, along with a sensitivity analysis showing how performance varies with learning rate. In plots and legends, the best performing learning rates are denoted by `*Our algorithm` and `*Hedge-QL`. For some plots, we also report results using a fixed learning rate. Finally, consistent with our theory, when referring to an instance of our algorithm with learning rate constant c_O , this corresponds to an implemented learning rate of $\eta = c_O \sqrt{(\log mNB)/T}$. Similarly, for Hedge-QL with constant c_H , the learning rate is set as $\eta = c_H \sqrt{(\log B)/T}/Y_{\max}$. We evaluate our algorithm over $c_O \in \{10^{i/4} \mid i = -8, \dots, 5\}$ and Hedge-QL over $c_H \in \{10^{i/4} \mid i = -4, \dots, 12\}$ (note that larger constants are needed for Hedge-QL to account for the normalization by $1/Y_{\max}$ which gives it a smaller effective learning rate).

4.4.1 Results

We present plots of omni-error over time, as well as total omni-error and total quantile loss for our algorithm and baseline ensembles. We also provide a sensitivity analysis with respect to the learning rate. Overall, the results show that our algorithm is comparable to or outperforms the baseline ensembles, with particularly strong performance on more difficult prediction tasks. Additional plots, including comparisons of quantile forecast trajectories and distributions of ranks of each model at each (horizon, location, time) tuple, are provided in Appendix C.5.2.

As a first evaluation, Figure 4.2 shows the omni-error over time for our algorithm, Hedge-QL, and the two existing ensembling models on the US-aggr data. We find that our algorithm outperforms the COVIDhub-Ens and JHUAPL-SLPEns ensembles across all time horizons with larger performance gaps occurring at longer time horizons where forecasting is more challenging. On the other hand, Hedge-QL performs similarly to our algorithm with the precise relative performance varying across horizons and learning rates. As a sensitivity check, this figure shows results for our algorithm obtained with both the best performing learning rate in hindsight and a fixed value of $c_O = 1$. This latter value was chosen as it minimizes the theoretical bound given in Theorem 4.3. Across all time horizons, forecasts from our algorithm with $c_O = 1$ perform as well as or better than the baseline ensembles.

To evaluate the generalizability of these results, the left panel of Figure 4.3 shows the differences in total omni-error (left column) and total quantile loss (right column) between our algorithm and each of the three comparators (rows). Similar to the results for US-aggr, we

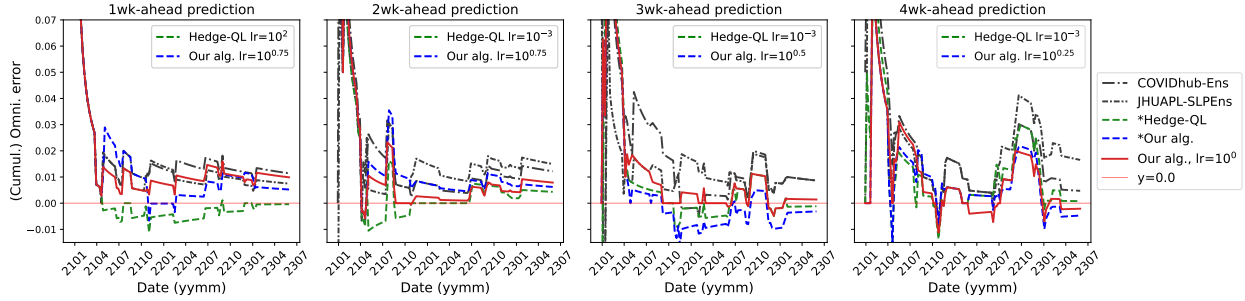


Figure 4.2: Omni-error (Eq. (4.10)) over time for 1- to 4-week-ahead forecasting horizons (left to right panels) on the US-agg data for **Our algorithm* (blue), **Hedge-QL* (green) and the COVIDhub-Ens and JHUAPL-SLPEns ensembles (both black). The values of c_H, c_O shown in the legend of each panel correspond to the learning rate which achieved the lowest total omni-error. The red line shows the results of our algorithm with a fixed learning rate constant $c_O = 1$. Lower values indicate better performance.

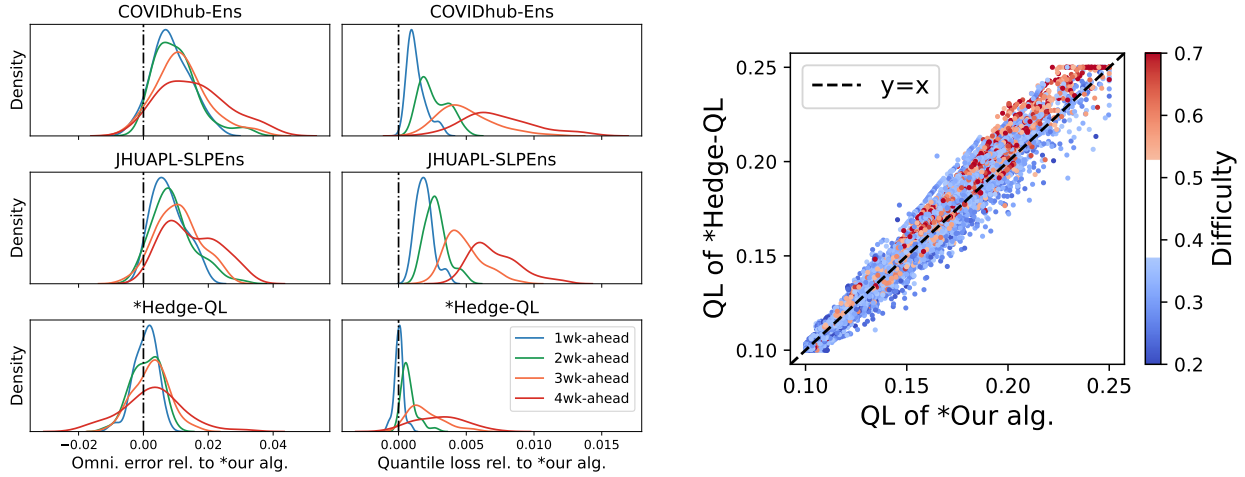


Figure 4.3: **Left:** State-level comparison of performance relative to **Our algorithm*. Rows show comparisons with three baseline ensembles. Each curve shows a Gaussian kernel density estimate of the difference in total omni-error (left column) or total quantile error (right column) across the 51 state-level time series for a given forecasting horizon. Values to the right of zero (vertical black dotted line) indicate better performance of our algorithm. **Right:** Comparison of instantaneous quantile loss realized by **Our algorithm* (x-axis) and **Hedge-QL* (y-axis). See text for detailed descriptions.

find that our algorithm consistently outperforms the COVIDhub-Ens and JHUAPL-SLPEns across all time horizons with greater gaps being observed for longer horizons. In comparison to Hedge-QL, we find that our algorithm has a small advantage in terms of the total omni-error and realizes a much bigger performance improvement in terms of the total quantile error especially at three and four week horizons.

The preceding results indicate that our algorithm shows the biggest performance gains on larger time horizons where forecasting is most difficult. To further investigate this trend, the right panel of Figure 4.3 displays the instantaneous quantile losses realized by our algorithm and Hedge-QL at each timestep. The color spectrum in this plot represents the difficulty of prediction at a given (horizon, location, time) tuple as measured by the average pinball loss of the base forecasters, $\frac{1}{Y_t} \frac{1}{BN} \sum_{n=1}^N \sum_{b=1}^B \rho_{\tau_n}(Y_t - f_b^n(t))$. We find that more difficult (darker red) points are predominantly located above the $y = x$ line indicating that our algorithm achieves lower quantile loss than Hedge-QL on harder prediction instances.

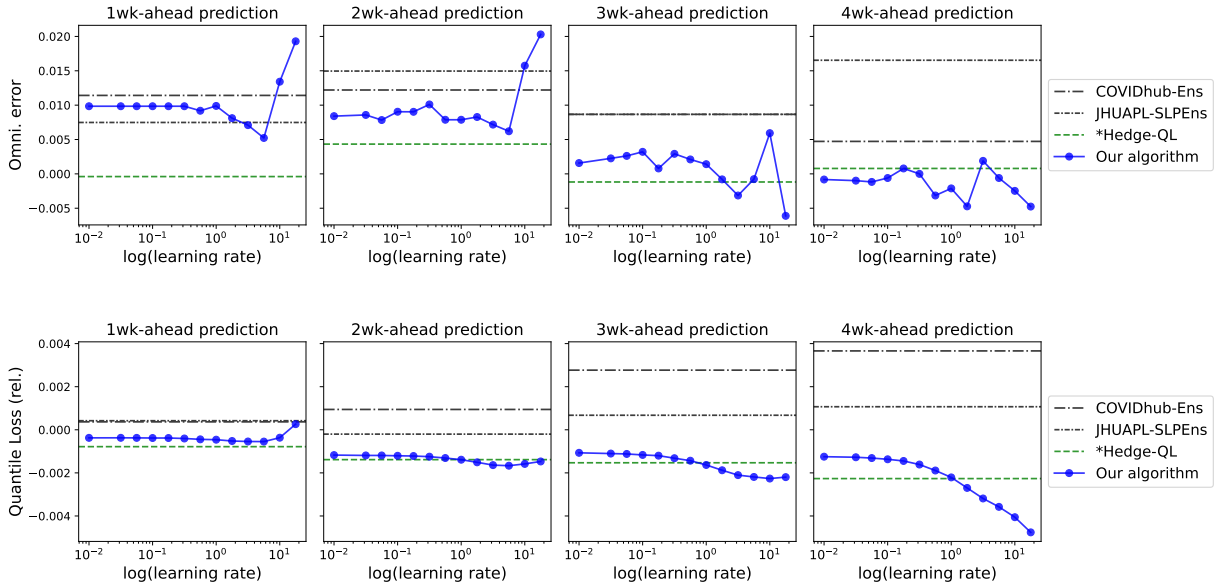


Figure 4.4: Sensitivity of our algorithm to the choice of learning rate. Each panel shows the performance of our algorithm as a function of the learning rate (blue curve) as measured by the total omni-error (top row) or total quantile loss relative to best base forecaster (bottom row) on the US-aggr data. Panels from left to right show results at 1- to 4-week time horizons, while horizontal lines indicate the performance of three baseline methods. Lower values indicate better performance.

We conclude this section with a sensitivity analysis evaluating how the performance of our algorithm varies with the learning rate. Figure 4.4 shows the realized total omni-error and total quantile loss realized by our algorithm for 1- to 4-week-ahead horizons on the US-aggr

data with learning rate constants $c_O \in \{10^{i/4} \mid i = -8, \dots, 5\}$. We see that across almost all horizons and learning rates our algorithm outperforms COVIDhub-Ens and JHUAPL-SLPEns and matches or improves on the performance of Hedge-QL. In general, we find that the total quantile loss varies more smoothly with respect to the learning rate than the total omni-error, although neither metric appears to be significantly sensitive to its value. A similar sensitivity plot for Hedge-QL is given in [Appendix C.5.2](#).

4.5 Discussion and conclusion

In this chapter we develop computationally and sample efficient omniprediction methods for continuous prediction targets. To make this problem tractable, our results focus on bounded proper losses for a finite set of quantile levels. While this setting captures a broad and practically relevant class of evaluation criteria, it does not cover unbounded losses or proper losses defined on the full predictive distribution. For example, the log-loss $\ell(f, y) = -\log f(y)$ falls into both categories and requires access to a forecasted density f . In general, quantile-based representations may be insufficient to evaluate such density-based proper losses, suggesting the need for new representations and algorithms beyond quantile aggregation. Exploring alternative aggregation criteria and determining the full scope of tractable omniprediction problems remain interesting directions for future work.

Bibliography

- [ABP⁺22] Taiga Abe, E. Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John Patrick Cunningham. Deep ensembles work, but are they necessary? In *Advances in Neural Information Processing Systems*, 2022.
- [ABPC22] Taiga Abe, E Kelly Buchanan, Geoff Pleiss, and John Patrick Cunningham. The best deep ensembles sacrifice predictive diversity. In *I Can’t Believe It’s Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*, 2022.
- [ALMV20] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *International Conference on Learning Representations (ICLR)*, 2020.
- [BCNM06] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.
- [BFSO84] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.
- [BGCT19] Davide Ballabio, Francesca Grisoni, Viviana Consonni, and Roberto Todeschini. Integrated qsar models to predict acute oral systemic toxicity. *Molecular Informatics*, 38(8-9):1800124, 2019.
- [BHMM19] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.
- [Bil13] Patrick Billingsley. *Convergence of probability measures*. John Wiley & Sons, 2nd edition, 2013.
- [BJRK22] Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, 35:19274–19289, 2022.

- [BL07] James Bennett and Stan Lanning. The netflix prize. 2007.
- [Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [BRT25] Erez Buchweitz, João Vitor Romano, and Ryan J. Tibshirani. Asymmetric penalties underlie proper loss functions in probabilistic forecasting, 2025.
- [CBIK⁺18] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.
- [CG16] 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.
- [CHW⁺22] Estee Y Cramer, Yuxin Huang, Yijin Wang, Evan L Ray, Matthew Cornell, Johannes Bracher, Andrea Brennen, Alvaro J Castro Rivadeneira, Aaron Gerding, Katie House, Dasuni Jayawardena, Abdul H Kanji, Ayush Khandelwal, Khoa Le, Jarad Niemi, Ariane Stark, Apurv Shah, Nutch Wattanachit, Martha W Zorn, Nicholas G Reich, and US COVID-19 Forecast Hub Consortium. The united states covid-19 forecast hub dataset. *Scientific Data*, 2022.
- [Con85] Marquis de Condorcet. Essay on the application of analysis to the probability of majority decisions. *Paris: Imprimerie Royale*, 1785.
- [CRL⁺22] Estee Y Cramer, Evan L Ray, Velma K Lopez, Johannes Bracher, Andrea Brennen, Alvaro J Castro Rivadeneira, Aaron Gerding, Tilmann Gneiting, Katie H House, Yuxin Huang, et al. Evaluation of individual and ensemble probabilistic forecasts of covid-19 mortality in the united states. *Proceedings of the National Academy of Sciences*, 119(15):e2113561119, 2022.
- [DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, June 2019.
- [Den12] Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [DG17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [EdB01] A. Engel and C. P. L. Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

- [EGJK16] Werner Ehm, Tilmann Gneiting, Alexander Jordan, and Fabian Krüger. Of quantiles and expectiles: consistent scoring functions, choquet representations and forecast rankings. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(3):505–562, 2016.
- [FHL19] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.
- [FKM⁺23] Rasool Fakoor, Taesup Kim, Jonas Mueller, Alexander J Smola, and Ryan J Tibshirani. Flexible model aggregation for quantile regression. *Journal of Machine Learning Research*, 24(162):1–45, 2023.
- [Fri01] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [FZ16] Tobias Fissler and Johanna F. Ziegel. Higher order elicibility and osband’s principle. *The Annals of Statistics*, 44(4), August 2016.
- [GJRR24] Sumegha Garg, Christopher Jung, Omer Reingold, and Aaron Roth. Oracle efficient online multicalibration and omniprediction. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2725–2792. SIAM, 2024.
- [GJS⁺20] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d’Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401, 2020.
- [GK14] Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(Volume 1, 2014):125–151, 2014.
- [GKR⁺22] Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, pages 79–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022.
- [GKR23] Parikshit Gopalan, Michael Kim, and Omer Reingold. Swap agnostic learning, or characterizing omniprediction via multicalibration. *Advances in Neural Information Processing Systems*, 36:39936–39956, 2023.

- [GLL⁺15] Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario March, and Jean-Francis Roy. Risk bounds for the majority vote: From a PAC-Bayesian analysis to a learning algorithm. *Journal of Machine Learning Research*, 16(26):787–860, 2015.
- [Gne11] Tilmann Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011.
- [GOR⁺24] Parikshit Gopalan, Princewill Okoroafor, Prasad Raghavendra, Abhishek Shetty, and Mihir Singhal. Omnipredictors for regression and the approximate rank of convex functions, 2024.
- [GR07] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [GRWG05] Tilmann Gneiting, Adrian E. Raftery, Anton H. Westveld, and Tom Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, 133(5):1098 – 1118, 2005.
- [GT25] Isaac Gibbs and Ryan J. Tibshirani. Sample-efficient omniprediction for proper losses, 2025.
- [Haz19] Elad Hazan. Introduction to online convex optimization. *CoRR*, abs/1909.05207, 2019.
- [HD19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.
- [HF16] Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.
- [HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.
- [How17] Andrew G Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [HTY26] Lunjia Hu, Kevin Tian, and Chutong Yang. Simultaneous blackwell approachability and applications to multiclass omniprediction, 2026.
- [HVD⁺15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

- [HZRS16] 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.
- [INLW21] Pavel Izmailov, Patrick Nicholson, Sanae Lotfi, and Andrew G Wilson. Dangers of Bayesian model averaging under covariate shift. *Advances in Neural Information Processing Systems*, 34:3309–3322, 2021.
- [IVHW21] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4629–4640. PMLR, 18–24 Jul 2021.
- [JNBK21] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J. Zico Kolter. Assessing generalization of SGD via disagreement, 2021.
- [Kal21] Olav Kallenberg. *Foundations of modern probability*. Springer, 3rd edition, 2021.
- [KB13] Vladimir S. Korolyuk and Yu V. Borovskich. *Theory of U-statistics*. Springer Science & Business Media, 2013.
- [KH⁺09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [KHTM24] Hyunsuk Kim, Liam Hodgkinson, Ryan Theisen, and Michael W Mahoney. How many classifiers do we need? *Advances in Neural Information Processing Systems*, 37:86458–86482, 2024.
- [Kor09] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.
- [Kot22] Vignesh Kothapalli. Neural collapse: A review on modelling principles and generalization. *arXiv preprint arXiv:2206.04041*, 2022.
- [LJGCW13] Kenneth C Lichtendahl Jr, Yael Grushka-Cockayne, and Robert L Winkler. Is it better to average probabilities or quantiles? *Management Science*, 59(7):1594–1611, 2013.
- [LMRR17] François Laviolette, Emilie Morvant, Liva Ralaivola, and Jean-Francis Roy. Risk upper bounds for general ensemble methods with an application to multiclass classification. *Neurocomputing*, 219:15–25, 2017.
- [LNPR22] Daniel Lee, Georgy Noarov, Mallesh Pai, and Aaron Roth. Online minimax multiobjective optimization: Multicalibrating and other applications. In *Advances in Neural Information Processing Systems*, 2022.

- [LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- [LRS25] Jiuyao Lu, Aaron Roth, and Mirah Shi. Sample efficient omniprediction and downstream swap regret for non-linear losses, 2025.
- [LW94] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- [McA98] David A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational Learning Theory*, pages 230–234, 1998.
- [MDP⁺11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, June 2011.
- [MIT⁺25] Cathal Mills, Nicholas J Irons, Joseph L-H Tsui, Sarah Sparrow, Luiz M Carvalho, Adam J Kucharski, Oliver Ratmann, Ben Lambert, Christl A Donnelly, and Moritz UG Kraemer. From metric to action: An evaluation framework to translate infectious disease forecasts into policy decisions. *medRxiv*, pages 2025–07, 2025.
- [MLIS20] Andrés Masegosa, Stephan Lorenzen, Christian Igel, and Yevgeny Seldin. Second order PAC-Bayesian bounds for the weighted majority vote. *Advances in Neural Information Processing Systems*, 33:5263–5273, 2020.
- [MM17] C. H. Martin and M. W. Mahoney. Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior. Technical Report Preprint: arXiv:1710.09553, 2017.
- [Mul24] Luke C Mullany. Jhuapl-slpHospEns. <https://github.com/lmullany/JHUAPL-SLPHospEns>, 2024.
- [NRRX25] Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making. In *Forty-second International Conference on Machine Learning*, 2025.
- [OCnM22] Luis A. Ortega, Rafael Cabañas, and Andres Masegosa. Diversity and generalization in neural network ensembles. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 11720–11743, 28–30 Mar 2022.

- [OFR⁺19] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.
- [OKK25] Princewill Okoroafor, Robert Kleinberg, and Michael P Kim. Near-optimal algorithms for omniprediction. *arXiv preprint arXiv:2501.17205*, 2025.
- [PVG⁺11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [QCHL87] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus. Inductive knowledge acquisition: A case study. In *Proceedings of the Second Australian Conference on Applications of Expert Systems*, page 137–156, USA, 1987. Addison-Wesley Longman Publishing Co., Inc.
- [Raf16] Adrian E. Raftery. Use and communication of probabilistic forecasts. *Statistical analysis and data mining*, 9(6):397–410, 2016.
- [RRSS19] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019.
- [Sae00] M. Saerens. Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks*, 11(6):1263–1271, 2000.
- [Sav71] Leonard J Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.
- [SWM⁺18] Danijel Schorlemmer, Maximilian J Werner, Warner Marzocchi, Thomas H Jordan, Yosihiko Ogata, David D Jackson, Sum Mak, David A Rhoades, Matthew C Gerstenberger, Naoshi Hirata, et al. The collaboratory for the study of earthquake predictability: Achievements and priorities. *Seismological Research Letters*, 89(4):1305–1313, 2018.
- [SYT⁺22] Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D’Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. The multiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*, 2022.

- [SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [TFF08] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.
- [Tho79] William Thomson. Eliciting production possibilities from a well-informed manager. *Journal of Economic Theory*, 20(3):360–380, 1979.
- [TKM21] Ryan Theisen, Jason Klusowski, and Michael Mahoney. Good classifiers are abundant in the interpolating regime. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3376–3384. PMLR, 13–15 Apr 2021.
- [TKY⁺24] Ryan Theisen, Hyunsuk Kim, Yaoqing Yang, Liam Hodgkinson, and Michael W. Mahoney. When are ensembles really effective? *Advances in Neural Information Processing Systems*, 36, 2024.
- [VBD⁺21] Stéphane Vannitsem, John Bjørnar Bremnes, Jonathan Demaeyer, Gavin R Evans, Jonathan Flowerdew, Stephan Hemri, Sebastian Lerch, Nigel Roberts, Susanne Theis, Aitor Atencia, et al. Statistical postprocessing for weather forecasts: Review, challenges, and avenues in a big data world. *Bulletin of the American Meteorological Society*, 102(3):E681–E699, 2021.
- [Vov90] Volodimir Vovk. Aggregating strategies. In *Proceedings of the Workshop on Computational Learning Theory*, 1990.
- [WSM⁺19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.
- [XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [YHT⁺21] Yaoqing Yang, Liam Hodgkinson, Ryan Theisen, Joe Zou, Joseph E Gonzalez, Kannan Ramchandran, and Michael W Mahoney. Taxonomizing local versus global structure in neural network loss landscapes. In *Advances in Neural Information Processing Systems*, volume 34, pages 18722–18733, 2021.

Appendix A

Appendix to Chapter 2

A.1 Proofs of main results

In this section, we provide proofs for our main results. Throughout the section, we denote $\mathbb{P}_{h,h'\sim\rho^2}$, $\mathbb{P}_{h\sim\rho}$, $\mathbb{E}_{h\sim\rho}$ by $\mathbb{P}_{h,h'}$, $\mathbb{P}_h$, $\mathbb{E}_h$, respectively. We also denote $W_\rho(X, Y)$ simply by W_ρ . We typically omit explicit dependence on the data distribution $\mathcal{D}$ when it is apparent from context.

A.1.1 Proof of Theorem 2.1

We first state and prove two lemmas that will be used in the proof of Theorem 2.1. Our first lemma states that majority-vote error $L_{\mathcal{D}}[h_{\text{MV}}]$ is upper bounded by probability of $W_\rho(X, Y)$ being large.

Lemma A.1. *There is the inequality $L_{\mathcal{D}}[h_{\text{MV}}] \leq \mathbb{P}_{\mathcal{D}}(W_\rho(X, Y) \geq 1/2)$, where $L_{\mathcal{D}}(h) = \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h(X) \neq Y)]$, $h_{\text{MV}}(\mathbf{x}) = \arg \max_j \mathbb{E}_h[\mathbb{1}(h(\mathbf{x}) = j)]$ and $W_\rho(X, Y) = \mathbb{E}_h[\mathbb{1}(h(X) \neq Y)]$.*

Proof. For given data point x , $W_\rho < 1/2$ implies that we are predicting the true label correctly more than half of the time. Thus, the majority vote classifier will correctly predict the label on the data point. $\square$

Our next lemma states a property of competent classifiers which plays a crucial role in the main proof.

Lemma A.2. *Under Assumption 2.1 (competence), for any increasing function h satisfying $h(0) = 0$,*

$$\mathbb{E}_{\mathcal{D}}[h(W_\rho)\mathbb{1}_{W_\rho < 1/2}] \geq \mathbb{E}_{\mathcal{D}}[h(\bar{W}_\rho)\mathbb{1}_{\bar{W}_\rho \leq 1/2}],$$

where $\bar{W}_\rho = 1 - W_\rho$.

Proof. For every $x \in [0, 1]$,

$$\begin{aligned}\mathbb{P}_{\mathcal{D}}(W_{\rho} \mathbf{1}_{W_{\rho} < 1/2} \geq x) &= \mathbb{P}_{\mathcal{D}}(W_{\rho} \in [x, 1/2)) \mathbf{1}_{x \leq 1/2}, \\ \mathbb{P}_{\mathcal{D}}(\bar{W}_{\rho} \mathbf{1}_{\bar{W}_{\rho} \leq 1/2} \geq x) &= \mathbb{P}_{\mathcal{D}}(\bar{W}_{\rho} \in [x, 1/2]) \mathbf{1}_{x \leq 1/2} = \mathbb{P}_{\mathcal{D}}(W_{\rho} \in [1/2, 1 - x]) \mathbf{1}_{x \leq 1/2}.\end{aligned}$$

From Assumption 2.1, this implies that $\mathbb{P}_{\mathcal{D}}(W_{\rho} \mathbf{1}_{W_{\rho} < 1/2} \geq x) \geq \mathbb{P}_{\mathcal{D}}(\bar{W}_{\rho} \mathbf{1}_{\bar{W}_{\rho} \leq 1/2} \geq x)$ for all $x \in [0, 1]$. Therefore, for any increasing function h satisfying $h(0) = 0$, since $h(x \mathbf{1}_{x \leq c}) = h(x) \mathbf{1}_{x \leq c}$,

$$\mathbb{P}_{\mathcal{D}}(h(W_{\rho}) \mathbf{1}_{W_{\rho} < 1/2} \geq x) \geq \mathbb{P}_{\mathcal{D}}(h(\bar{W}_{\rho}) \mathbf{1}_{\bar{W}_{\rho} \leq 1/2} \geq x).$$

As W_{ρ} is non-negative, the equality $\mathbb{E}X = \int_0^{\infty} \mathbb{P}(X \geq x)dx$ concludes the proof. $\square$

With these two lemmas, we now provide the proof of Theorem 2.1.

Proof of Theorem 2.1. From Lemma A.1 and the relation $\mathbb{E}_h[L_{\mathcal{D}}(h)] = \mathbb{E}_{\mathcal{D}}[W_{\rho}]$ (Fubini's theorem), it suffices to show that $\mathbb{P}_{\mathcal{D}}(W_{\rho} \geq 1/2) \leq \mathbb{E}_{\mathcal{D}}[W_{\rho}]$. To do so, observe

$$\mathbb{E}_{\mathcal{D}}[(W_{\rho} - 1) \mathbf{1}_{W_{\rho} \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[\bar{W}_{\rho} \mathbf{1}_{\bar{W}_{\rho} \leq 1/2}] = \mathbb{E}_{\mathcal{D}}[(W_{\rho} - 1) \mathbf{1}_{W_{\rho} \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[(1 - W_{\rho}) \mathbf{1}_{W_{\rho} \geq 1/2}] = 0.$$

Applying Lemma A.2 with $h(x) = x$,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[W_{\rho}] - \mathbb{P}_{\mathcal{D}}(W_{\rho} \geq 1/2) &\geq \mathbb{E}_{\mathcal{D}}[(W_{\rho} - 1) \mathbf{1}_{W_{\rho} \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[W_{\rho} \mathbf{1}_{W_{\rho} < 1/2}] \\ &\geq \mathbb{E}_{\mathcal{D}}[(W_{\rho} - 1) \mathbf{1}_{W_{\rho} \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[\bar{W}_{\rho} \mathbf{1}_{\bar{W}_{\rho} \leq 1/2}] = 0.\end{aligned}$$

which proves

$$L_{\mathcal{D}}[h_{\text{MV}}] \underset{\text{Lemma A.1}}{\leq} \mathbb{P}_{\mathcal{D}}(W_{\rho} \geq 1/2) \leq \mathbb{E}_{\mathcal{D}}[W_{\rho}] = \mathbb{E}_h[L_{\mathcal{D}}(h)]. \quad (\text{A.1})$$

This implies $\text{EIR} \geq 0$. $\square$

A.1.2 Proof of Theorem 2.2

A.1.2.1 Lower bound of EIR

To prove the lower bound, we first define the tandem loss, as used in [MLIS20].

Definition A.1 (Tandem loss). *Define the tandem loss to be $L(h, h') = \mathbb{E}_{\mathcal{D}}[\mathbf{1}(h(X) \neq Y) \mathbf{1}(h'(X) \neq Y)]$.*

We also rely on the following lemma, which appears as Lemma 2 in [MLIS20]. It provides the connection between the average error rate for each data point, W_{ρ} and the tandem loss, $L(h, h')$.

Lemma A.3. *The equality $\mathbb{E}_{\mathcal{D}}[W_{\rho}^2] = \mathbb{E}_{h, h'}[L(h, h')]$ holds.*

We first state and prove the following lemma, which provides an upper bound on the tandem loss.

Lemma A.4. *For the K -class problem,*

$$\mathbb{E}_{h,h'}[L(h, h')] \leq \frac{2(K-1)}{K} \left(\mathbb{E}_h[L(h)] - \frac{1}{2} \mathbb{E}_{h,h'}[D(h, h')] \right).$$

Proof. We denote $\mathbb{P}_h(h(X) \neq Y)$ by $\bar{h}_Y(X)$. Note that $\mathbb{E}_{\mathcal{D}}(1 - \bar{h}_Y(X)) = \mathbb{E}_h[L(h)]$ and

$$\begin{aligned} \mathbb{E}_{h,h'}[L(h, h')] &= \mathbb{E}_{\mathcal{D}}[\mathbb{P}_h(h(X) \neq Y) \mathbb{P}_{h'}(h'(X) \neq Y)] \\ &= \mathbb{E}_{\mathcal{D}}[(1 - \bar{h}_Y(X))^2]. \end{aligned}$$

Then we get

$$\begin{aligned} \mathbb{E}_{h,h'}[L(h, h')] &= \mathbb{E}_{\mathcal{D}}[(1 - \bar{h}_Y(X))^2] \\ &= 1 - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)] - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1 - \bar{h}_Y(X))] \\ &= \mathbb{E}_h[L(h)] - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1 - \bar{h}_Y(X))]. \end{aligned}$$

Now we will derive a lower bound of the second term. Since

$$\mathbb{E}_{h,h'}[\mathbb{1}(h(X) \neq h'(X))] = \sum_j \bar{h}_j(X)(1 - \bar{h}_j(X)),$$

it follows that

$$\bar{h}_Y(X)(1 - \bar{h}_Y(X)) = \mathbb{E}_{h,h'}[\mathbb{1}(h(X) \neq h'(X))] - \sum_{j \neq Y} \bar{h}_j(X)(1 - \bar{h}_j(X)).$$

By maximizing $\sum_{j \neq Y} \bar{h}_j(X)(1 - \bar{h}_j(X))$ subject to $\sum_{j \neq Y} \bar{h}_j(X) = 1 - \bar{h}_Y(X)$, we get $\bar{h}_j(X) = \frac{1 - \bar{h}_Y(X)}{K-1}$, which yields the upper bound

$$\sum_{j \neq Y} \bar{h}_j(X)(1 - \bar{h}_j(X)) \leq \frac{K-2}{K-1}(1 - \bar{h}_Y(X)) + \frac{1}{K-1} \bar{h}_Y(X)(1 - \bar{h}_Y(X)).$$

It follows that

$$\mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1 - \bar{h}_Y(X))] \geq \frac{K-1}{K} \mathbb{E}_{h,h'}[D(h, h')] - \frac{K-2}{K} \mathbb{E}_h[L(h)],$$

and thus that

$$\begin{aligned} \mathbb{E}_{h,h'}[L(h, h')] &= \mathbb{E}_h[L(h)] - \mathbb{E}_{\mathcal{D}}[\bar{h}_Y(X)(1 - \bar{h}_Y(X))] \\ &\leq \mathbb{E}_h[L(h)] - \left(\frac{K-1}{K} \mathbb{E}_{h,h'}[D(h, h')] - \frac{K-2}{K} \mathbb{E}_h[L(h)] \right) \\ &= \frac{2(K-1)}{K} \left(\mathbb{E}_h[L(h)] - \frac{1}{2} \mathbb{E}_{h,h'}[D(h, h')] \right). \end{aligned}$$

□

We now provide the proof for the lower bound of EIR in Theorem 2.2.

Proof. We first claim that $\mathbb{P}_{\mathcal{D}}(W_{\rho} \geq 1/2) \leq 2 \mathbb{E}_{\mathcal{D}}[W_{\rho}^2]$. Then, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[(2W_{\rho}^2 - 1)\mathbb{1}_{W_{\rho} \geq 1/2}] &= \mathbb{E}_{\mathcal{D}}[(2(1 - \bar{W}_{\rho})^2 - 1)\mathbb{1}_{\bar{W}_{\rho} \leq 1/2}] \\ &= \mathbb{E}_{\mathcal{D}}[(1 - 4\bar{W}_{\rho} + 2\bar{W}_{\rho}^2)\mathbb{1}_{\bar{W}_{\rho} \leq 1/2}], \end{aligned}$$

where $\bar{W}_{\rho} = 1 - W_{\rho}$. Therefore,

$$\mathbb{E}_{\mathcal{D}}[(2W_{\rho}^2 - 1)\mathbb{1}_{W_{\rho} \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[2\bar{W}_{\rho}^2\mathbb{1}_{\bar{W}_{\rho} \leq 1/2}] = \mathbb{E}_{\mathcal{D}}[(1 - 4\bar{W}_{\rho} + 4\bar{W}_{\rho}^2)\mathbb{1}_{\bar{W}_{\rho} \leq 1/2}] \geq 0.$$

Now we apply Lemma A.2 with $h(x) = 2x^2$, to obtain

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[2W_{\rho}^2] - \mathbb{P}_{\mathcal{D}}(W_{\rho} \geq 1/2) &\geq \mathbb{E}_{\mathcal{D}}[(2W_{\rho}^2 - 1)\mathbb{1}_{W_{\rho} \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[2W_{\rho}^2\mathbb{1}_{W_{\rho} < 1/2}] \\ &\geq \mathbb{E}_{\mathcal{D}}[(2W_{\rho}^2 - 1)\mathbb{1}_{W_{\rho} \geq 1/2}] + \mathbb{E}_{\mathcal{D}}[2\bar{W}_{\rho}^2\mathbb{1}_{\bar{W}_{\rho} \leq 1/2}] \geq 0, \end{aligned} \quad (\text{A.2})$$

which proves the claim, $\mathbb{P}_{\mathcal{D}}(W_{\rho} \geq 1/2) \leq 2 \mathbb{E}_{\mathcal{D}}[W_{\rho}^2]$.

Now we put the claim together with Lemmas A.1, A.3, and A.4 to conclude the proof.

$$\begin{aligned} L(h_{\text{MV}}) &\stackrel{\text{Lemma A.1}}{\leq} \mathbb{P}_{\mathcal{D}}(W_{\rho} \geq 1/2) \leq 2 \mathbb{E}_{\mathcal{D}}[W_{\rho}^2] \stackrel{\text{Lemma A.3}}{=} 2 \mathbb{E}_{h,h'}[L(h, h')] \\ &\stackrel{\text{Lemma A.4}}{\leq} \frac{4(K-1)}{K} \left(\mathbb{E}_h[L(h)] - \frac{1}{2} \mathbb{E}_{h,h'}[D(h, h')] \right) \end{aligned} \quad (\text{A.3})$$

Rearranging the terms, we obtain

$$\mathbb{E}_h[L_{\mathcal{D}}(h)] - L(h_{\text{MV}}) \geq \frac{2(K-1)}{K} \mathbb{E}_{h,h'}[D(h, h')] - \frac{3K-4}{K} \mathbb{E}_h[L(h)]. \quad (\text{A.4})$$

Dividing the both terms by $\mathbb{E}_h[L(h)]$ gives the lower bound $\frac{2(K-1)}{K} \text{DER} - \frac{3K-4}{K}$. $\square$

A.1.2.2 Upper bound of EIR

We denote $\mathbb{P}_h(h(X) \neq Y)$ by $\bar{h}_Y(X)$. We have

$$\mathbb{E}_h[L(h)] - L(h_{\text{MV}}) = \mathbb{E}_{h,\mathcal{D}}[\mathbb{1}(h(X) \neq Y) - \mathbb{1}(h_{\text{MV}}(X) \neq Y)].$$

Now

$$\begin{aligned} \mathbb{1}(h(X) \neq Y) - \mathbb{1}(h_{\text{MV}}(X) \neq Y) &= \mathbb{1}(h_{\text{MV}}(X) = Y) - \mathbb{1}(h(X) = Y) \\ &= \mathbb{1}(h(X) \neq h_{\text{MV}}(X)) (\mathbb{1}(h_{\text{MV}}(X) = Y) - \mathbb{1}(h(X) = Y)) \\ &\leq \mathbb{1}(h(X) \neq h_{\text{MV}}(X)). \end{aligned}$$

Now notice $\mathbb{E}_{h,\mathcal{D}}[1(h(X) \neq h_{\text{MV}}(X))] = 1 - \mathbb{E}_{\mathcal{D}}[\max_k \bar{h}_k(X)]$. Moreover, by Hölder's inequality,

$$\|\bar{\mathbf{h}}(X)\|_2^2 \leq \max_k \bar{h}_k(X),$$

and so

$$\begin{aligned} \mathbb{E}_h[L(h)] - L(h_{\text{MV}}) &\leq 1 - \mathbb{E}_{\mathcal{D}}[\max_k \bar{h}_k(X)] \\ &\leq 1 - \mathbb{E}_{\mathcal{D}}[\|\bar{\mathbf{h}}(X)\|_2^2] = \mathbb{E}_{h,h'}[D(h, h')]. \end{aligned} \tag{A.5}$$

Dividing the both terms by $\mathbb{E}_h[L(h)]$ gives the upper bound DER.

A.1.3 Upper and lower bounds on the error rate of the majority vote classifier

We now present upper and lower bound on the majority vote classifier that follow from the bounds in Theorem 2.1 and 2.2, and compare them with existing bounds in the literature.

Theorem A.1. *For any competent ensemble ρ of K -class classifiers, the majority vote error rate satisfies*

$$\begin{aligned} L(h_{\text{MV}}) &\leq \min \left\{ \frac{4(K-1)}{K} \left(\mathbb{E}_{h \sim \rho}[L(h)] - \frac{1}{2} \mathbb{E}_{h,h' \sim \rho}[D(h, h')] \right), \mathbb{E}_{h \sim \rho}[L(h)] \right\} \\ L(h_{\text{MV}}) &\geq \mathbb{E}_{h \sim \rho}[L(h)] - \mathbb{E}_{h,h' \sim \rho}[D(h, h')]. \end{aligned}$$

Proof. The upper bound follows from inequality (A.1) and (A.3). The lower bound follows from inequality (A.5). $\square$

We have already discussed that the bound $L(h_{\text{MV}}) \leq \mathbb{E}[L(h)]$ represents an improvement by a factor of 2 over the naive first-order bound (2.1). Here, we further compare the bound

$$L(h_{\text{MV}}) \leq \frac{4(K-1)}{K} \left(\mathbb{E}_{h \sim \rho}[L(h)] - \frac{1}{2} \mathbb{E}_{h,h' \sim \rho}[D(h, h')] \right) \tag{A.6}$$

to other known results in the literature. The closest in form is a bound specialized to binary case from [MLIS20], which gives

$$L(h_{\text{MV}}) \leq 4\mathbb{E}_{h \sim \rho}[L(h)] - 2\mathbb{E}_{h,h' \sim \rho}[D(h, h')]. \tag{A.7}$$

Note that plugging in $K = 2$ to (A.6), we obtain the bound $2\mathbb{E}_{h \sim \rho}[L(h)] - \mathbb{E}_{h,h' \sim \rho}[D(h, h')]$, immediately improving on (A.7) by a factor of 2 (interestingly, the same factor that we save on the first-order bound). Hence, provided the competence assumption holds, our bound is a direct improvement on this bound, and furthermore generalizes directly to the K -class setting.

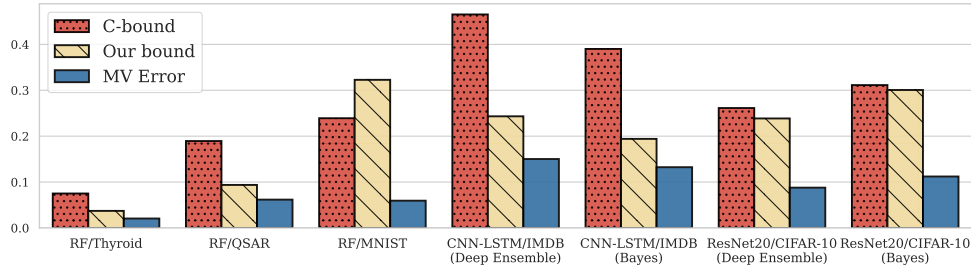


Figure A.1: **Our bound (A.6) versus the multi-class C-bound (A.8).**

To our knowledge, the sharpest known upper bound on the majority-vote classifier is the general form of the C-bound given in [LMRR17], which states, provided $\mathbb{E}[M_\rho(X, Y)] > 0$,

$$L(h_{\text{MV}}) \leq 1 - \frac{\mathbb{E}[M_\rho(X, Y)]^2}{\mathbb{E}[M_\rho^2(X, Y)]}, \quad (\text{A.8})$$

where $M_\rho(X, Y) = \mathbb{E}_{h \sim \rho}[\mathbb{1}(h(X) = Y)] - \max_{j \neq Y} \mathbb{E}_{h \sim \rho}[\mathbb{1}(h(X) = j)]$ is called the *margin*. Unfortunately, the use of the margin function makes direct analytical comparison to our bound difficult. However, the bounds can be compared empirically, where the relevant quantities are estimated on hold-out data. In Figure A.1, we compare the value of our bound against the value of the multi-class C-bound, on tasks for which we have verified the competence assumption holds. We find that in all but one case (random forests with MNIST), our bound is superior empirically, sometimes significantly. Interestingly, we observe that our bound does particularly well on tasks with only a few classes. This behavior might be attributed to the constant $\frac{4(K-1)}{K}$ in the upper bound (A.6) which increases as the number of classes K grows.

A.1.4 Exact characterization of the EIR in the regression case

Here we briefly derive an analogous expression for the EIR in the regression case, as mentioned in Remark 2.2. In this case models are regressors f drawn from a distribution ρ , and the natural ensemble model is $\bar{f}(\mathbf{x}) = \mathbb{E}_{f \sim \rho}[f(\mathbf{x})]$. Here we use as a loss function the squared error $L(f) = \mathbb{E}_{X, Y}[(f(X) - Y)^2]$. We now derive a bias-variance-like expression relating the the average error rate to the ensemble error rate.

Lemma A.5. *For any distribution ρ over regressors, the average error $\mathbb{E}_{f \sim \rho}[L(f)]$ satisfies*

$$\mathbb{E}_{f \sim \rho}[L(f)] = L(\bar{f}) + \mathbb{E}_X[\text{Var}_{f \sim \rho}[f(X)]].$$

Proof. The proof is a simple bias-variance decomposition, taken with respect to the random-

ness in models f drawn from ρ . We have

$$\begin{aligned}
\mathbb{E}_f[L(f)] &\stackrel{(\star)}{=} \mathbb{E}_{X,Y}[\mathbb{E}_f[(Y - f(X))^2]] \\
&= \mathbb{E}_{X,Y} \left[\mathbb{E}_f[(Y - \bar{f}(X) + \bar{f}(X) - f(X))^2] \right] \\
&= \mathbb{E}_{X,Y} \left[\mathbb{E}_f[(Y - \bar{f}(X))^2] + 2\mathbb{E}_f[(Y - \bar{f}(X))(\bar{f}(X) - f(X))] + \mathbb{E}_f[(\bar{f}(X) - f(X))^2] \right] \\
&= L(\bar{f}) + \mathbb{E}_X[\text{Var}_{f \sim \rho}[f(X)]] + 2\mathbb{E}_f[(Y - \bar{f}(X))(\bar{f}(X) - f(X))],
\end{aligned}$$

where in $(\star)$ we use Fubini's theorem to exchange integrals. To complete the proof, we claim that the last term is equal to zero. Indeed,

$$\begin{aligned}
\mathbb{E}_f[(Y - \bar{f}(X))(\bar{f}(X) - f(X))] &= Y\bar{f}(X) - Y\mathbb{E}_f[f(X)] - \bar{f}^2(X) + \bar{f}(X)\mathbb{E}_f[f(X)] \\
&= Y\bar{f}(X) + Y\bar{f}(X) - \bar{f}^2(X) + \bar{f}^2(X) = 0.
\end{aligned}$$

□

Now if we analogously define the EIR in the regression case as

$$\text{EIR} = \frac{\mathbb{E}_f[L(f)] - L(\bar{f})}{\mathbb{E}_f[L(f)]}$$

then we obtain directly the following corollary analogous to Theorem 2.2.

Corollary A.1. *The EIR in the regression case satisfies*

$$\text{EIR} = \frac{\mathbb{E}_f[L(f)] - L(\bar{f})}{\mathbb{E}_f[L(f)]} = \frac{\mathbb{E}_X[\text{Var}_{f \sim \rho}[f(X)]]}{\mathbb{E}_f[L(f)]}.$$

Note here the variance-average error ratio $\mathbb{E}_X[\text{Var}_{f \sim \rho}[f(X)]]/\mathbb{E}_f[L(f)]$ plays the role of the DER in the classification case.

A.2 Additional empirical results

A.2.1 Experimental details

Bagged random feature classifiers. We consider ensembles of random ReLU feature classifiers, constructed as follows. For each classifier, we draw a random matrix $\mathbf{U} \in \mathbb{R}^{N \times d}$, whose rows $\mathbf{u}_j$ are drawn from the uniform distribution on the sphere $\mathbb{S}^{d-1}$. For a given input $\mathbf{x} \in \mathbb{R}^d$, we compute the feature $\mathbf{z}(\mathbf{x}) = \sigma(\mathbf{U}\mathbf{x})$ where $\sigma(t) = \max(t, 0)$ is the ReLU function. We then fit a multi-class logistic regression model in `scikit-learn` [PVG⁺11] using these (random) features. To form an ensemble of these classifiers, we additionally perform bagging, by sampling a different set of size n with replacement from the training set of size n , independently for each individual classifier. Thus, each classifier is subject to two different types of randomness: the randomness from the sampling of the feature matrix $\mathbf{U}$; and the randomness from the bootstrapping of the training data. For the models shown in the competence plot in Figure 2.1, we use 500 random features and $M = 100$ classifiers.

Random forests. We consider random forest (RF) models as implemented in `scikit-learn` [PVG⁺11], each made up of 20 individual decision trees. We vary the maximum number of leaf nodes in each tree to construct models with varying performance. For the single-ensemble results presented, we use the default parameters implemented in `scikit-learn`. For the random forests, we use a small version of the MNIST dataset with 5000 randomly selected training examples (500 from each of the 10 classes). We also use two binary classification datasets retrieved from the UCI repository [DG17]: the QSAR oral toxicity dataset (7.2k train, 1.8k test examples, 1024 features) [BGCT19]; and the Thyroid disease dataset (2.5k train, 633 test examples, 21 features) [QCHL87]. For the models shown in the competence plot in Figure 2.1, we use the default settings of the random forest implementation in `scikit-learn`.

Deep ensembles. We consider four different architectures for our deep ensembles. We use a standard ResNet18 models [HZRS16] trained on the CIFAR-10 dataset [KH⁺09], using 100 epochs of SGD with momentum 0.9, weight decay of 5×10^{-4} and a learning rate of 0.1, while varying the batch size and width hyper-parameters. We report results from two variants of this empirical evaluation: one in which we employ learning rate decay (by dropping the learning rate to 0.01 after 75 epochs); and another in which we disable learning rate decay. For each setting, we train 5 models from independent initialization to form the respective ensembles. We also evaluate these models on two out-of-distribution databases: CIFAR-10.1 and CIFAR-10-C [RRSS19, HD19] (the latter is itself comprised of 19 different datasets employing various types of data corruption). Finally, we evaluate deep ensembles of 25 standard BERT models [DCLT19], provided with the paper [SYT⁺22], fine-tuned on the GLUE classification tasks [WSM⁺19].

Bayesian ensembles. For the Bayesian ensembles used in Chapter 2, we consider samples provided in [IVHW21], obtained via large-scale sampling from a Bayesian posterior using Hamiltonian Monte Carlo. To our knowledge, these samples are the most precisely representative of a theoretical Bayesian neural network posterior publicly available. In particular, we use samples on the CIFAR-10 datasets with a ResNet20 architecture, and the IMDB dataset on the CNN-LSTM architecture. We defer to the original paper [IVHW21] for additional details.

A.2.2 More competence plots

In this section, we provide additional empirical results.

To further verify that the competence assumption holds broadly in practice, here we include several more examples of competence plots for experiments presented in the main text.

ResNet18 on CIFAR-10 OOD variants. In Figures A.2 and A.3, we plot competence plots for the ResNet18 ensembles on the CIFAR-10, CIFAR-10.1 and a subset of the CIFAR-10-C datasets [RRSS19, HD19]. We find that the competence assumption holds across all examples.

Fine-tuned BERT models. In Figure A.4, we provide competence plots for the BERT/GLUE fine-tuning tasks. For the RTE, CoLA, MNLI, QQP and QNLI tasks, we find that the competence assumption holds. However, we find two examples here where it does not: the MRPC and SST-2 tasks, although the extent to which the assumption is violated is minor. Since these are particularly small datasets, this may also be a product of noise from low sample size.

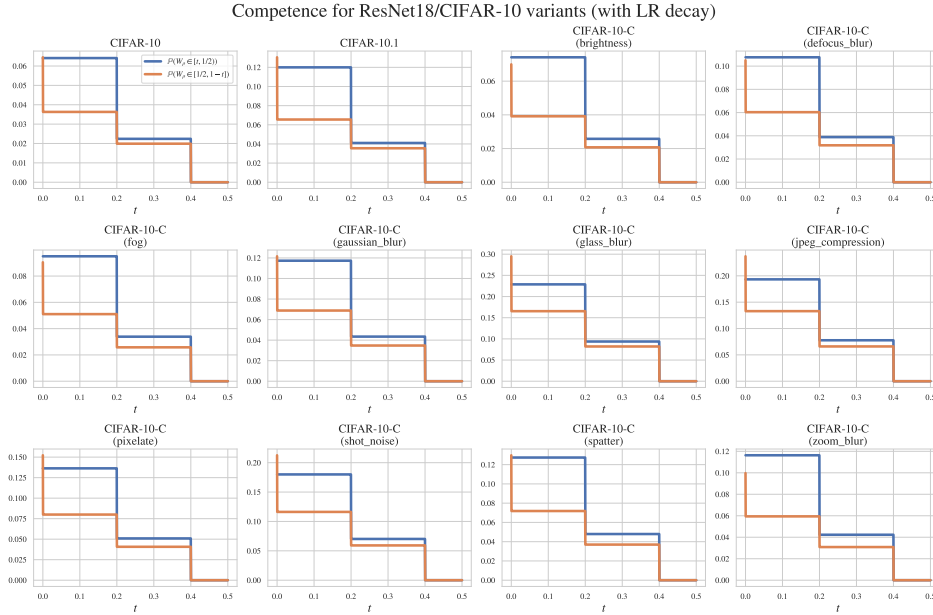


Figure A.2: **Competence for ResNet18/CIFAR-10 variants (models with learning rate decay).** We observe that the competence assumption holds across all tasks.

A.3 Pathological ensembles where majority vote fails

In this section, we provide two pathological examples of ensembles that makes the “first-order” upper bound tight. In particular, the second example shows that positive margin condition, i.e., $\mathbb{E}[M_\rho(X, Y)] > 0$ where $M_\rho(X, Y) = \mathbb{E}_{h \sim \rho}[\mathbf{1}(h(X) = Y)] - \max_{j \neq Y} \mathbb{E}_{h \sim \rho}[\mathbf{1}(h(X) = j)]$, from existing literature is not enough to rule out pathological cases. Recall that the first-order bound introduced in Section 2.2.2 is the following:

$$0 \leq L(h_{\text{MV}}) \leq \mathbb{P}(W_\rho \geq 1/2) \leq 2 \mathbb{E}[W_\rho] = 2 \mathbb{E}_{h \sim \rho}[L(h)].$$

Example A.1 (The first-order upper bound is tight). *Consider a classification problem with two classes. For given $\epsilon > 0$, suppose slightly less than half, $0.5 - \epsilon$, fraction of classifiers are the perfect classifier, correctly classifying test data with probability 1, and the other $0.5 + \epsilon$ fraction of classifiers are completely wrong, incorrectly predicting on test data with probability*

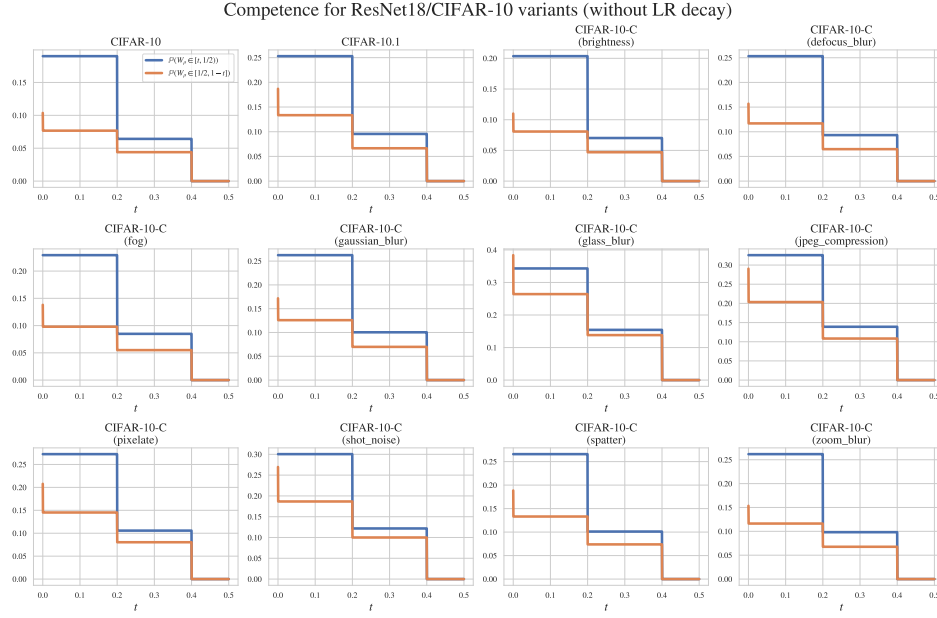


Figure A.3: **Competence for ResNet18/CIFAR-10 variants (models without learning rate decay)**. We observe that the competence assumption holds across all tasks.

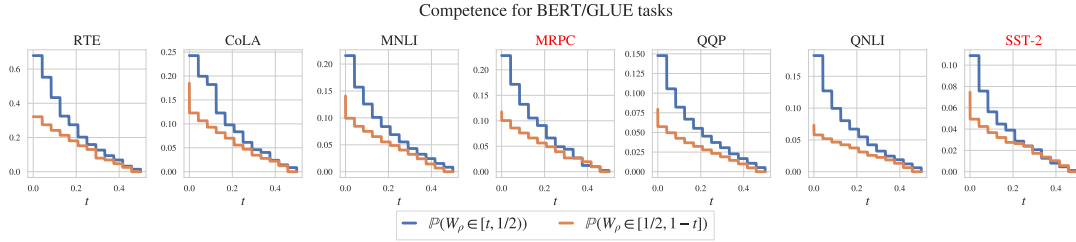
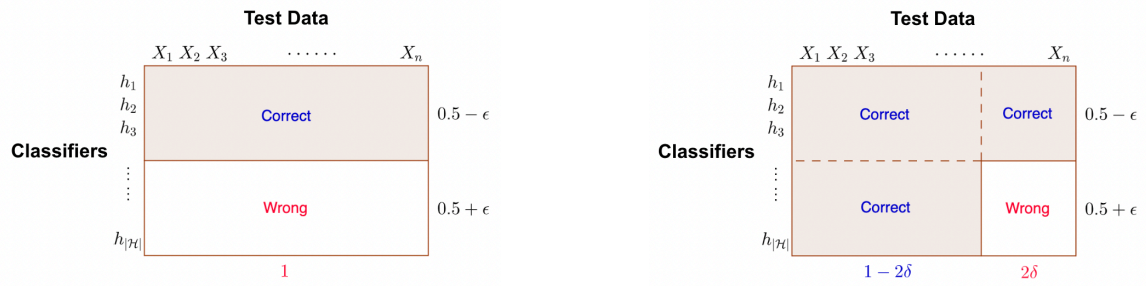


Figure A.4: **Competence for BERT/GLUE fine-tuning tasks**. The competence assumption holds for the RTE, CoLA, MNLI, QQP and QNLI tasks, though interestingly, we find that the competence assumption is (to a small degree) violated for two of the tasks: MRPC and SST-2.

1. With this composition of classifiers, the average error rate is $0.5 + \epsilon$ and the majority vote error rate is 1. Taking $\epsilon \rightarrow 0$ concludes that the first-order upper bound (2.1) is tight. A visual illustration of the composition of classifiers is given in Figure A.5a.

The condition $\mathbb{E}[M_\rho(X, Y)] > 0$ rules out the ensemble described in Example A.1. Nonetheless, the first-order bound $2\mathbb{E}[L(h)]$ is tight *even when* $\mathbb{E}[M_\rho(X, Y)] > 0$ is satisfied, as we show with the following example.



(a) Composition of classifiers in Example A.1

(b) Composition of classifiers in Example A.2

Figure A.5: **Illustration of the composition of classifiers given in Examples A.1 and A.2.** On each plot, the area of the white box equals to the average test error rate. On Figure A.5a, the majority vote error rate is 1, while the average test error rate is $0.5 + \epsilon$. On Figure A.5b, the majority vote error rate is 2δ (Rightmost 2δ test data) while the average test error rate is $\delta(1 + 2\epsilon)$. The margin of each composition of classifiers is $2\epsilon \rightarrow 0$ and $1 - 2\delta(1 + 2\epsilon) > 0$, respectively.

Example A.2 (The first-order upper bound is tight even when the margin is large). We again consider a classification problem with two classes. For given $\epsilon > 0$, as in Example A.1, slightly less than half, $0.5 - \epsilon$, fraction of classifiers are the perfect classifier. All of the other $0.5 + \epsilon$ fraction of classifiers, on the contrary, now correctly predict on the same $1 - 2\delta$ fraction of the test data and incorrectly predict on the other 2δ fraction of the test data. With this composition of classifiers, the majority vote error rate is 2δ even when the average error rate is $\delta(1 + 2\epsilon)$. In addition, unlike the composition of classifiers in Example A.1, the margin of which is 2ϵ , the margin of the new composition of classifiers is $1 - 2\delta(1 + 2\epsilon)$, which can be any value smaller than 1. Taking $\epsilon \rightarrow 0$ concludes that the first-order upper bound (2.1) is also tight when the margin is arbitrarily high. A visual illustration of the composition of classifiers is given in Figure A.5b.

Appendix B

Appendix to Chapter 3

B.1 More discussion on competence

In this section, we delve more into the competence condition that was introduced in [TKY⁺24]. We explore in which cases the competence condition might not work and how to overcome these issues. We discuss a few milder versions of competence that are enough for bounds (3.5) and (3.6) to hold. Then we discuss how to check whether these weaker competence conditions hold in practice, with or without a separate validation set. We start by formally stating the original competence condition.

Definition B.1 (Competence, [TKY⁺24]). *The ensemble ρ is **competent** if for every $0 \leq t \leq 1/2$,*

$$\mathbb{P}_{\mathcal{D}}(W_{\rho} \in [t, 1/2)) \geq \mathbb{P}_{\mathcal{D}}(W_{\rho} \in [1/2, 1 - t]). \quad (\text{B.1})$$

B.1.1 Cases when competence fails

One tricky part in the definition of competence is that it requires inequality (B.1) to hold for **every** $0 \leq t \leq 1/2$. In case $t = 1/2$, the inequality becomes

$$0 \geq \mathbb{P}_{\mathcal{D}}(W_{\rho} = 1/2).$$

This is not a significant issue in the case that ρ is a continuous distribution over classifiers, e.g., a Bayes posterior or a distribution over a parametric family h_{θ} , as $\{W_{\rho} = 1/2\}$ would be a measure-zero set. In the case that ρ is a discrete distribution over finite number of classifiers, however, $\mathbb{P}_{\mathcal{D}}(W_{\rho} = 1/2)$ is likely to be a positive quantity, in which case it can violate the competence condition.

That being said, $\{(x, y) \mid W_{\rho}(x, y) = 1/2\}$ represent tricky data points that deserves separate attention. This event can be divided into two cases: 1) all the classifiers that incorrectly made a prediction output the same label; or 2) incorrect predictions consist of multiple labels

so that the majority vote outputs the true label. Among these two possibilities, the first case is troublesome. We denote such data points by $\text{TIE}(\rho, \mathcal{D})$:

$$\begin{aligned} \text{TIE}(\rho, \mathcal{D}) := \\ \{(x, y) \mid \mathbb{P}_\rho(\mathbb{1}(h(x)=j)) = \mathbb{P}_\rho(\mathbb{1}(h(x)=y)) = 1/2 \text{ for true label } y \text{ and an incorrect label } j\}. \end{aligned}$$

In this case, the true label and an incorrect label are chosen by exactly the same ρ -weights of classifiers. An easy way to resolve this issue is to slightly tweak the weights. For instance, if ρ is an equally weighted sum of two classifiers, we can change each of their weights to be $(1/2 + \epsilon, 1/2 - \epsilon)$, instead of $(1/2, 1/2)$. This change may seem manipulative, but it corresponds to a deterministic tie-breaking rule which prioritizes one classifier over the other, which is a commonly used tie-breaking rule.

Definition B.2 (Tie-free ensemble). *An ensemble is **tie-free** if $\mathbb{P}_{\mathcal{D}}(\text{TIE}(\rho, \mathcal{D})) = 0$.*

Proposition B.1. *An ensemble with a deterministic tie-breaking rule is tie-free.*

With such tweak to make the set $\text{TIE}(\rho, \mathcal{D})$ to be an empty set or a measure-zero set, we present a slightly milder condition that is enough for the bounds (3.5) and (3.6) to still hold.

Definition B.3 (Semi-competence). *The ensemble ρ is **semi-competent** if for every $0 \leq t < 1/2$,*

$$P(W_\rho \in [t, 1/2]) \geq \mathbb{P}(W_\rho \in (1/2, 1 - t]). \quad (\text{B.2})$$

Note that inequality (B.2) is a strictly weaker condition than inequality (B.1), and hence competence implies semi-competence. The converse is not true. An ensemble is semi-competent even if the point-wise error $W_\rho(X, Y) = 1/2$ on every data points, but such an ensemble is not competent.

Theorem B.1. *For a tie-free ensemble and semi-competent ensemble ρ , $L(h_\rho^{\text{MV}}) \leq \mathbb{E}_\rho[L(h)]$ and*

$$L(h_\rho^{\text{MV}}) \leq \frac{4(K-1)}{K} \left(\mathbb{E}_\rho[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right)$$

holds in K -class classification setting.

We provide the proof as a separate subsection below.

B.1.2 Proof of Theorem B.1 and Proposition 3.1

We start with the following lemma, which is a semi-competence version of Lemma 2 from [TKY⁺24].

Lemma B.1. *For a semi-competent ensemble ρ and any increasing function g satisfying $g(0) = 0$,*

$$\mathbb{E}_{\mathcal{D}}[g(W_{\rho})\mathbf{1}_{W_{\rho} \leq 1/2}] \geq \mathbb{E}_{\mathcal{D}}[g(\widetilde{W}_{\rho})\mathbf{1}_{\widetilde{W}_{\rho} < 1/2}],$$

where $\widetilde{W}_{\rho} = 1 - W_{\rho}$.

Proof. For every $x \in [0, 1]$, it holds that

$$\begin{aligned} \mathbb{P}_{\mathcal{D}}(W_{\rho}\mathbf{1}_{W_{\rho} \leq 1/2} \geq x) &= \mathbb{P}_{\mathcal{D}}(W_{\rho} \in [x, 1/2]) \mathbf{1}_{x \leq 1/2}, \\ \mathbb{P}_{\mathcal{D}}(\widetilde{W}_{\rho}\mathbf{1}_{\widetilde{W}_{\rho} < 1/2} \geq x) &= \mathbb{P}_{\mathcal{D}}(\widetilde{W}_{\rho} \in [x, 1/2)) \mathbf{1}_{x \leq 1/2} = \mathbb{P}_{\mathcal{D}}(W_{\rho} \in (1/2, 1 - x]) \mathbf{1}_{x \leq 1/2}. \end{aligned}$$

From the definition of semi-competence, this implies that

$$\mathbb{P}_{\mathcal{D}}(W_{\rho}\mathbf{1}_{W_{\rho} \leq 1/2} \geq x) \geq \mathbb{P}_{\mathcal{D}}(\widetilde{W}_{\rho}\mathbf{1}_{\widetilde{W}_{\rho} < 1/2} \geq x),$$

for every $x \in [0, 1]$. Using the fact that $g(x\mathbf{1}_{x \leq c}) = g(x)\mathbf{1}_{x \leq c}$ for any increasing function g with $g(0) = 0$, we obtain

$$\mathbb{P}_{\mathcal{D}}(h(W_{\rho})\mathbf{1}_{W_{\rho} \leq 1/2} \geq x) \geq \mathbb{P}_{\mathcal{D}}(h(\widetilde{W}_{\rho})\mathbf{1}_{\widetilde{W}_{\rho} < 1/2} \geq x).$$

Putting these together with a well-known equality $\mathbb{E}X = \int_0^{\infty} \mathbb{P}(X \geq x)dx$ for a non-negative random variable X proves the lemma. $\square$

Now we use Lemma B.1 and Theorem 3.2 to prove Theorem B.1.

Proof of Theorem B.1. Applying Lemma B.1 with $g(x) = 2x^2$ gives,

$$\mathbb{E}_{\mathcal{D}}[2W_{\rho}^2\mathbf{1}_{W_{\rho} \leq 1/2}] \geq \mathbb{E}_{\mathcal{D}}[2\widetilde{W}_{\rho}^2\mathbf{1}_{\widetilde{W}_{\rho} < 1/2}] = \mathbb{E}_{\mathcal{D}}[(2 - 4W_{\rho} + 2W_{\rho}^2)\mathbf{1}_{W_{\rho} > 1/2}]. \quad (\text{B.3})$$

Putting this together with the following decomposition of $\mathbb{E}_{\mathcal{D}}[2W_{\rho}^2]$ shows that the ensemble ρ is 2-polarized:

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[2W_{\rho}^2] &\geq \mathbb{E}_{\mathcal{D}}[2W_{\rho}^2\mathbf{1}_{W_{\rho} > 1/2}] + \mathbb{E}_{\mathcal{D}}[2W_{\rho}^2\mathbf{1}_{W_{\rho} \leq 1/2}] \\ &\stackrel{(\text{B.3})}{\geq} \mathbb{E}_{\mathcal{D}}[2W_{\rho}^2\mathbf{1}_{W_{\rho} > 1/2}] + \mathbb{E}_{\mathcal{D}}[(2 - 4W_{\rho} + 2W_{\rho}^2)\mathbf{1}_{W_{\rho} > 1/2}] \\ &\geq \mathbb{E}_{\mathcal{D}}[(1 - 2W_{\rho})^2\mathbf{1}_{W_{\rho} > 1/2}] + \mathbb{P}_{\mathcal{D}}(W_{\rho} > 1/2) \\ &\geq \mathbb{P}_{\mathcal{D}}(W_{\rho} > 1/2). \end{aligned} \quad (\text{B.4})$$

Therefore, applying Theorem 3.2 with constant $\eta = 2$ concludes the proof. $\square$

We also state the following proof of Proposition 3.1 for completeness.

Proof of Proposition 3.1. Inequality (B.4) with Lemma B.3 proves the proposition. $\square$

B.1.3 Example that the bound (3.6) is tight

Here, we provide a combination of $(\rho, \mathcal{D})$ of which $L(h_\rho^{\text{MV}})$ is arbitrarily close to the bound. Consider, for each feature x , that exactly $(1 - \epsilon)$ fraction of classifiers predict the correct label, and that the remaining ϵ fraction of classifiers predict a wrong label. In this case, $L(h_\rho^{\text{MV}}) = 0$, $\mathbb{E}_\rho[L(h)] = \epsilon$, and $\mathbb{E}_{\rho^2}[D(h, h')] = 2\epsilon(1 - \epsilon)$. Hence, the upper bound (3.6) is $\frac{4(K-1)}{K}\epsilon^2$, which can be arbitrarily close to 0.

B.2 Proofs of main results

In this section, we provide proofs for our main results.

B.2.1 Proof of Theorem 3.1

We start with the following lemma which shows the concentration of a linear combination of W_ρ^2 and $\mathbf{1}(W_\rho > 1/2)$.

Lemma B.2. *For sampled data points $\{(X_i, Y_i)\}_{i=1}^m \sim \mathcal{D}$, define $Z_2 := \sum_{i=1}^m W_\rho^2(X_i, Y_i)$ and $Z_0 := \sum_{i=1}^m \mathbf{1}(W_\rho(X_i, Y_i) > 1/2)$. The ensemble ρ is η -polarized with probability at least $1 - \delta$ if*

$$\frac{1}{m}(\eta Z_2 - Z_0) > \sqrt{\frac{\max\{\frac{3\eta}{4}, 1\}}{2m} \log \frac{1}{\delta}}. \quad (\text{B.5})$$

Proof. Let $Z_{2i} = W_\rho^2(X_i, Y_i)$ and $Z_{0i} = \mathbf{1}(W_\rho(X_i, Y_i) > 1/2)$. Observe that $\eta Z_{2i} - Z_{0i}$ always takes a value between $[\frac{\eta}{4} - 1, \max\{\frac{\eta}{4}, \eta - 1\}]$ since $W_\rho(X_i, Y_i) \in [0, 1]$. This implies that $\eta Z_{2i} - Z_{0i}$ s are i.i.d. sub-Gaussian random variable with parameter $\sigma = \max\{\frac{3\eta}{4}, 1\}/2$.

By letting $A_2 = \mathbb{E}[\eta W_\rho^2 - \mathbf{1}(W_\rho \geq 1/2)]$ and using the Hoeffding's inequality, we obtain

$$\frac{1}{m}(\eta Z_2 - Z_0) - A_2 \leq \sqrt{\frac{\max\{\frac{3\eta}{4}, 1\}}{2m} \log \frac{1}{\delta}}$$

with probability at least $1 - \delta$.

Therefore, ρ is η -polarized with probability at least $1 - \delta$ if

$$\frac{1}{m}(\eta Z_2 - Z_0) > \sqrt{\frac{\max\{\frac{3\eta}{4}, 1\}}{2m} \log \frac{1}{\delta}}.$$

□

Now we use Lemma B.2 to prove Theorem 3.1.

Proof of Theorem 3.1. Observe that $S = \frac{1}{m}Z_2$, $P = \frac{1}{m}Z_0$, and thus $\frac{1}{m}(\eta Z_2 - Z_0) = \eta S - P$. For $\eta \geq \frac{4}{3}$, the lower bound in Lemma B.2 is simply $\sqrt{\frac{3\eta}{8m} \log \frac{1}{\delta}}$, and the inequality (B.5) can be viewed as a quadratic inequality in terms of $\sqrt{\eta}$. From quadratic formula, we know that

$$\text{if } \sqrt{\eta} > \frac{\sqrt{\frac{3\eta}{8m} \log \frac{1}{\delta}} + \sqrt{\frac{3\eta}{8m} \log \frac{1}{\delta} + 4SP}}{2S}, \quad \text{then } \eta S - P - \sqrt{\frac{3\eta}{8m} \log \frac{1}{\delta}} > 0.$$

Putting this together with Lemma B.2 proves the theorem:

$$\begin{aligned} \eta &\geq \max \left\{ \frac{4}{3}, \left(\frac{\sqrt{\frac{3}{8m} \log \frac{1}{\delta}} + \sqrt{\frac{3}{8m} \log \frac{1}{\delta} + 4SP}}{2S} \right)^2 \right\} \\ &\Rightarrow \eta S - P > \sqrt{\frac{3\eta}{8m} \log \frac{1}{\delta}} \xrightarrow{\text{Lemma B.2}} \rho \text{ is } \eta\text{-polarized w.p. } 1 - \delta, \end{aligned} \quad (\text{B.6})$$

and thus the polarization η_ρ , the smallest η such that ρ is η -polarized, is upper bounded by the right hand side of inequality (B.6). $\square$

B.2.2 Proof of Theorem 3.2

We start by proving the following lemma which relates the error rate of the majority vote, $L(h_\rho^{\text{MV}})$, with the point-wise error rate, W_ρ , using Markov's inequality. In general, $L(h_\rho^{\text{MV}}) \leq \mathbb{P}_\mathcal{D}(W_\rho \geq 1/2)$ is true for any ensemble ρ . We prove a tighter version of this. The difference between the two can be non-negligible when dealing with an ensemble with finite number of classifiers. Refer to Appendix B.1.1 and Definition B.2 for more details regarding this difference and tie-free ensembles.

Lemma B.3. *For a tie-free ensemble ρ , we have the inequality $L(h_\rho^{\text{MV}}) \leq \mathbb{P}_\mathcal{D}(W_\rho > 1/2)$.*

Proof. For given feature x , $W_\rho \leq 1/2$ implies that more than or exactly ρ -weighted half of the classifiers outputs the true label. Since the ensemble ρ is tie-free, h_ρ^{MV} outputs the true label if $W_\rho \leq 1/2$. Therefore, $\{(x, y) \mid W_\rho(x, y) \leq 1/2\} \subset \{(x, y) \mid h_\rho^{\text{MV}}(x) = y\}$. Applying $\mathbb{P}_\mathcal{D}$ on the both sides proves the lemma. $\square$

The following lemma appears as Lemma 2 in [MLIS20]. This lemma draws the connection between the point-wise error rate, W_ρ and the tandem loss, $\mathbb{E}_{\rho^2}[L(h, h')]$.

Lemma B.4. *The equality $\mathbb{E}_\mathcal{D}[W_\rho^2] = \mathbb{E}_{\rho^2}[L(h, h')]$ holds.*

The next lemma appears as Lemma 4 in [TKY⁺24]. This lemma provides an upper bound on the tandem loss, $\mathbb{E}_{\rho^2}[L(h, h')]$, in terms of the average error rate, $\mathbb{E}_\rho[L(h)]$, and the average disagreement, $\mathbb{E}_{\rho^2}[D(h, h')]$.

Lemma B.5. *For the K -class problem,*

$$\mathbb{E}_{\rho^2}[L(h, h')] \leq \frac{2(K-1)}{K} \left(\mathbb{E}_{\rho}[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right).$$

Now we use these results to prove Theorem 3.2.

Proof of Theorem 3.2. Putting Lemmas B.3, B.4, and B.5 and the definition of the polarization together proves the theorem:

$$\begin{aligned} L(h_{\rho}^{\text{MV}}) &\stackrel{\text{Lemma B.3}}{\leq} \mathbb{P}_{\mathcal{D}}(W_{\rho} > 1/2) \stackrel{\text{polarization}}{\leq} \eta_{\rho} \mathbb{E}_{\mathcal{D}}[W_{\rho}^2] \\ &\stackrel{\text{Lemma B.4}}{=} \eta_{\rho} \mathbb{E}_{\rho^2}[L(h, h')] \stackrel{\text{Lemma B.5}}{=} \frac{2\eta_{\rho}(K-1)}{K} \left(\mathbb{E}_h[L(h)] - \frac{1}{2} \mathbb{E}_{h, h'}[D(h, h')] \right). \end{aligned}$$

□

B.2.3 Proof of Theorem 3.3

We start with a lemma which is a corollary of *Newton's inequality*.

Lemma B.6. *For any collection of probabilities $p_1, \dots, p_n$, the following inequality holds.*

$$\sum_{1 \leq i < j \leq n} p_i p_j \leq \frac{n-1}{2n} \left(\sum_{i=1}^n p_i \right)^2.$$

Proof. Newton's inequality states that

$$\frac{e_2}{\binom{n}{2}} \leq \left(\frac{e_1}{n} \right)^2 \quad \text{where} \quad e_1 = \sum_{i=1}^n p_i \quad \text{and} \quad e_2 = \sum_{1 \leq i < j \leq n} p_i p_j.$$

Rearranging the terms gives the lemma. □

Now we use this and the previous lemmas to prove Theorem 3.3.

Proof of Theorem 3.3. From Lemma B.3, Lemma B.4, and the definition of η -polarized ensemble, we have the following relationship between $L(h_{\rho}^{\text{MV}})$ and $\mathbb{E}_{\rho^2}[L(h, h')]$:

$$L(h_{\rho}^{\text{MV}}) \stackrel{\text{Lem. B.3}}{\leq} \mathbb{P}_{\mathcal{D}}(W_{\rho} > 1/2) \stackrel{\eta\text{-polarized}}{\leq} \eta \mathbb{E}_{\mathcal{D}}[W_{\rho}^2] \stackrel{\text{Lemma B.4}}{=} \eta \mathbb{E}_{\rho^2}[L(h, h')]. \quad (\text{B.7})$$

From this, it suffices to prove that $h_{\alpha} \mathbb{E}_{\rho^2}[L(h, h')]$ is smaller than the upper bound in the theorem. First, observe the following decomposition of $\mathbb{E}_{\rho^2}[L(h, h')]$:

$$\mathbb{E}_{\rho^2}[L(h, h')] = \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y)^2] = \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y) - \mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) = Y)]. \quad (\text{B.8})$$

For any predictor mapping into K classes, let y denote the true label for an input x . Now we derive a lower bound of $\mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) = Y)$ using the following decomposition of $\mathbb{P}_{\rho^2}(h(x) \neq h'(x))$:

$$\begin{aligned} & \frac{1}{2} \mathbb{P}_{\rho^2}(h(x) \neq h'(x)) \\ &= \mathbb{P}_{\rho^2}(h(x) \neq y, h'(x) = y) + \sum_{\substack{i \notin A(x) \\ j \in A(x) \setminus \{y\}}} p_i p_j + \sum_{\substack{i, j \in A(x) \setminus \{y\} \\ i < j}} p_i p_j + \sum_{\substack{i, j \notin A(x) \\ i < j}} p_i p_j, \end{aligned}$$

where $p_i := p_i(x) = \mathbb{P}_{\rho}(h(x) = i)$. We let $\Delta_x := \mathbb{P}_{\rho}(h(x) \notin A(x))$ and apply Lemma B.6 to the last two terms:

$$\begin{aligned} & \frac{1}{2} \mathbb{P}_{\rho^2}(h(x) \neq h'(x)) \\ &= \mathbb{P}_{\rho^2}(h(x) \neq y, h'(x) = y) + \Delta_x(1 - p_Y - \Delta_x) + \sum_{\substack{i, j \in A(x) \setminus \{y\} \\ i < j}} p_i p_j + \sum_{\substack{i, j \notin A(x) \\ i < j}} p_i p_j \\ &\stackrel{\text{Lemma B.6}}{\leq} \mathbb{P}_{\rho^2}(h(x) \neq y, h'(x) = y) + \Delta_x(1 - p_Y - \Delta_x) \\ &\quad + \frac{M-2}{2(M-1)}(1 - p_Y - \Delta_x)^2 + \frac{K-M-1}{2(K-M)}\Delta_x^2. \end{aligned}$$

Rearranging the terms and plugging $1 - p_Y = \mathbb{P}_{\rho}(h(x) \neq y)$ gives

$$\begin{aligned} & \mathbb{P}_{\rho^2}(h(x) \neq y, h'(x) = y) \\ &\geq \frac{1}{2} \mathbb{P}_{\rho^2}(h(x) \neq h'(x)) - \frac{\Delta_x}{M-1} \mathbb{P}_{\rho}(h(x) \neq y) - \frac{M-2}{2(M-1)} \mathbb{P}_{\rho}(h(x) \neq y)^2 \\ &\quad + \frac{K-1}{2(K-M)(M-1)} \Delta_x^2 \\ &\geq \frac{1}{2} \mathbb{P}_{\rho^2}(h(x) \neq h'(x)) - \frac{\Delta_x}{M-1} \mathbb{P}_{\rho}(h(x) \neq y) - \frac{M-2}{2(M-1)} \mathbb{P}_{\rho}(h(x) \neq y)^2 \\ &\geq \frac{1}{2} \mathbb{P}_{\rho^2}(h(x) \neq h'(x)) - \frac{\Delta}{M-1} \mathbb{P}_{\rho}(h(x) \neq y) - \frac{M-2}{2(M-1)} \mathbb{P}_{\rho}(h(x) \neq y)^2, \end{aligned}$$

where the last inequality comes from the condition $\Delta_x := \mathbb{P}_{\rho}(h(x) \notin A(x)) \leq \Delta$. Putting this together with the equality (B.8) gives

$$\begin{aligned} \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y)^2] &\leq \left(1 + \frac{\Delta}{M-1}\right) \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y)] - \frac{1}{2} \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho^2}(h(X) \neq h'(X))] \\ &\quad + \frac{M-2}{2(M-1)} \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y)^2], \end{aligned}$$

which implies

$$\begin{aligned}\mathbb{E}_{\rho^2}[L(h, h')] &= \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y)^2] \\ &\leq \frac{2(M-1)}{M} \left[\left(1 + \frac{\Delta}{M-1}\right) \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y)] - \frac{1}{2} \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho^2}(h(X) \neq h'(X))] \right] \\ &= \frac{2(M-1)}{M} \left[\left(1 + \frac{\Delta}{M-1}\right) \mathbb{E}_{\rho}[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right].\end{aligned}$$

Combining this with inequality (B.7) concludes the proof. $\square$

B.2.4 Proof of Theorem 3.4 and Corollary 3.2

First, we prove Theorem 3.4 by decomposing the point-wise disagreement between constituent classifiers.

Proof of Theorem 3.4. The following decomposition of $\mathbb{P}_{\rho^2}(h(x) \neq h'(x))$ holds:

$$\frac{1}{2} \mathbb{P}_{\rho^2}(h(x) \neq h'(x)) = \mathbb{P}_{\rho^2}(h(x) \neq y, h'(x) = y) + \frac{1}{2} \mathbb{P}_{\rho^2}(h(x) \neq y, h'(x) = y, h(x) \neq h'(x)).$$

Applying $\mathbb{E}_{\mathcal{D}}$ to both sides and using the given condition (3.11), we obtain,

$$\frac{1}{2} \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho^2}(h(X) \neq h'(X))] \leq \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) = Y)] + \varepsilon \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y)].$$

The left hand side equals $\frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')]$, and the second term on the right hand side is simply $\varepsilon \mathbb{E}_{\rho}[L(h)]$. Hence, the inequality above can be rephrased as follows:

$$\frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] - \varepsilon \mathbb{E}_{\rho}[L(h)] \leq \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) = Y)]. \quad (\text{B.9})$$

Putting this together with the inequality (B.7) and the equality (B.8), gives

$$\begin{aligned}L(h_{\rho}^{\text{MV}}) &\stackrel{\text{Ineq. B.7}}{\leq} \eta \mathbb{E}_{\rho^2}[L(h, h')] \stackrel{\text{Eq. B.8}}{=} \eta \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho}(h(X) \neq Y) - \mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) = Y)] \\ &= \eta [\mathbb{E}_{\rho}[L(h)] - \mathbb{E}_{\mathcal{D}} [\mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) = Y)]] \\ &\stackrel{\text{Ineq. B.9}}{\leq} \eta \left[(1 + \varepsilon) \mathbb{E}_{\rho}[L(h)] - \frac{1}{2} \mathbb{E}_{\rho^2}[D(h, h')] \right].\end{aligned}$$

$\square$

Next, we use Lemma B.6 to prove Corollary 3.2.

Proof of Corollary 3.2. Let $p_i := \mathbb{P}_\rho(h(x) = i)$ for $i \in [K]$, and let $y = K$ be the true label, without loss of generality. Then, we observe

$$\mathbb{P}_{\rho^2}(h(X) \neq Y, h'(X) \neq Y, h(X) \neq h'(X)) = \sum_{\substack{i,j=1 \\ i \neq j}}^{K-1} p_i p_j \quad \text{and} \quad \mathbb{P}_\rho(h(X) \neq Y) = \sum_{i=1}^{K-1} p_i.$$

Lemma B.6 gives us the following:

$$\frac{\sum_{1 \leq i \neq j \leq K-1} p_i p_j}{2 \sum_{1 \leq i \leq K-1} p_i} \leq \frac{\sum_{1 \leq i \neq j \leq K-1} p_i p_j}{2(\sum_{1 \leq i \leq K-1} p_i)^2} \leq \frac{\sum_{1 \leq i < j \leq K-1} p_i p_j}{(\sum_{1 \leq i \leq K-1} p_i)^2} \stackrel{\text{Lemma B.6}}{\leq} \frac{K-2}{2(K-1)},$$

where the first inequality used the fact that $\sum_{i=1}^{K-1} p_i \leq 1$. Thus, $\varepsilon = \frac{K-2}{2(K-1)}$ satisfies the condition (3.11), and the result follows from Theorem 3.4. $\square$

B.2.5 Proof of Theorem 3.5

Proof. Let $\Phi(h_i, h_j) = \mathbb{P}_\mathcal{D}(h_i(X) \neq h_j(X))$. We observe that

$$\begin{aligned} \frac{N^2}{N(N-1)} \mathbb{E}_{(h,h') \sim \rho_N^2} [D(h, h')] &= \frac{1}{N(N-1)} \sum_{i,j=1}^N \mathbb{P}_\mathcal{D}(h_i(X) \neq h_j(X)) \\ &= \frac{1}{N(N-1)} \sum_{i,j=1}^N \Phi(h_i, h_j) \stackrel{\substack{\Phi: \text{symmetric} \\ \Phi(h_i, h_i) = 0}}{=} \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} \Phi(h_i, h_j) =: U_N, \end{aligned} \quad (\text{B.10})$$

which is a U -statistic with the kernel function Φ . Let $\Phi_0 = \mathbb{E}_{(h,h') \sim \rho^2} \Phi(h, h')$.

The invariance principle of U -statistics (Theorem B.2 in Appendix B.2.6) states that the process $\xi_N = (\xi_N(t), t \in [0, 1])$, defined by $\xi_N(\frac{k}{N}) = \frac{k}{2\sqrt{N\sigma_1^2}}(U_k - \Phi_0)$ and $\xi_N(t) = \xi_N(\frac{[Nt]}{N})$, converges weakly to a standard Wiener process in $\mathcal{D}[0, 1]$ as $N \rightarrow \infty$, since $\sigma_1^2 = \text{Var}_{h \sim \rho} \mathbb{E}_{h' \sim \rho} \Phi(h, h')$. Therefore, U_N converges in probability as $N \rightarrow \infty$ to $D_\infty := \Phi_0$.

Letting $Z_N = \sigma_1 \xi_N(1) = \frac{\sqrt{N}}{2}(U_N - D_\infty)$, we can express U_N as $U_N = D_\infty + \frac{2}{\sqrt{N}} Z_N$, with $\mathbb{E} Z_N = 0$ and $\text{Var} Z_N \rightarrow \sigma_1^2$. Since $\frac{\sqrt{t}}{\sigma_1} Z_{[Nt]} = \sqrt{\frac{Nt}{[Nt]}} \xi_N(\frac{[Nt]}{N}) = \sqrt{\frac{Nt}{[Nt]}} \xi_N(t)$, it follows by Slutsky's Theorem that $\{\frac{\sqrt{t}}{\sigma_1} Z_{[Nt]}\}_{t \in [0, 1]}$ converges weakly to a standard Wiener process in $\mathcal{D}[0, 1]$ as $N \rightarrow \infty$. $\square$

B.2.6 Invariance principle of U -statistics

In this subsection, we state the invariance principle of U -statistics, which plays a main role in the proof of Theorem 3.5. We note that this is a special case of an approximation of random walks (Theorem 23.14 in [Kal21]) combined with functional central limit theorem (Donsker's theorem). Here, $\mathcal{D}[0, 1]$ is the Skorokhod space on $[0, 1]$, which is the space of all real-valued right-continuous functions on $[0, 1]$ equipped with the Skorokhod metric/topology (see Section 14 in [Bil13]).

Theorem B.2 (Theorem 5.2.1 in [KB13]). Define a U -statistic $U_k = \binom{k}{2}^{-1} \sum_{1 \leq i < j \leq k} \Phi(h_i, h_j)$, the expectation of the kernel Φ as $\Phi_0 = \mathbb{E}_{(h, h') \sim \rho^2} \Phi(h, h')$ and the first-coordinate variance $\sigma_1^2 = \text{Var}_{h \sim \rho}(g_1(h))$, where $g_1(h) = \mathbb{E}_{h' \sim \rho} \Phi(h', h)$. Let $\xi_n = (\xi_n(t), t \in [0, 1])$, where

$$\xi_n\left(\frac{k}{n}\right) = \frac{k(U_k - \Phi_0)}{2\sqrt{n\sigma_1^2}} \quad \text{for } k = 0, 1, \dots, n-1,$$

and $\xi_n(t) = \xi_n([nt]/n)$, with $[x]$ denoting the greatest integer less than or equal to x . Then, ξ_n converges weakly in $\mathcal{D}[0, 1]$ to a standard Wiener process as $n \rightarrow \infty$.

B.3 Experiment details

In this section, we provide additional details on our empirical results.

B.3.1 Trained classifiers

On CIFAR-10 [KH⁺09] train set with size 50,000, the following models were trained with 100 epochs, learning rate starting with 0.05. For models trained with learning rate decay, we used learning rate 0.005 after epoch 50, and used 0.0005 after epoch 75. For following models, 5 classifiers are trained for each hyperparameter combination. Five classifiers differ in weight initialization and vary due to the randomized batches used during training.

- ResNet18, every combination (width, batch size) of
 - Width: 4, 8, 16, 32, 64, 128
 - Batch size: 16, 128, 256, 1024, with learning rate decay
 - Additional batch size of 64, 180, 364 for without learning rate decay
- ResNet50, ResNet101, every combination (width, batch size) of
 - Width: 8, 16
 - Batch size: 64, 256, without learning rate decay
- VGG11, every combination (width, batch size) of
 - Width: 16, 64
 - Batch size: 64, 256, without learning rate decay
- DenseNet40, every combination (width, batch size) of
 - Width: 5, 12, 40
 - Batch size: 64, 256, without learning rate decay

For models in Figure 3.4, more than 5 classifiers were trained. The classifiers differ in weight initialization and vary due to the randomized batches used during training.

- ResNet18 on CIFAR-10, width 16 and batch size 64 without learning rate decay (20 classifiers)

The models below are trained with learning rate 0.05, momentum 0.9 and weight decay $5e-4$ with cosine annealing.

- MobileNet on MNIST, batch size 128 (10 classifiers)
- ResNet18 on FMNIST, width 48 and batch size 128 (10 classifiers)
- ResNet18 on KMNIST, every combination of widths and batch sizes below (8 classifiers each)
 - Width: 48, 64
 - Batch size: 32, 64, 128

B.3.2 Majority vote and tie-free

For an ensemble with N classifiers, we generated N uniformly-distributed random numbers $e_1, \dots, e_N \in [0, 0.0001]$. Then used $(\frac{1}{N} + e_1, \dots, \frac{1}{N} + e_N)$ after normalization as weights for each classifier. This guarantees the ensemble to be tie-free.

Appendix C

Appendix to Chapter 4

C.1 Proofs of main results

C.1.1 Proof of Lemma 4.1

Proof. Let $\theta_{i^*} \in \{\theta_i\}_{i=1}^m$ be the closest value from θ , where the ties are broken by rounding up. Observe that $\mathbb{1}(\theta < a) = \mathbb{1}(\theta_{i^*} < a)$ for $a \in \{\frac{j}{m}\}_{j=0}^m$. This gives

$$\begin{aligned}\ell_{\tau,\theta}(x, y) &= (\mathbb{1}(y < x) - \tau)(\mathbb{1}(\theta < x) - \mathbb{1}(\theta < y)) \\ &= (\mathbb{1}(y < x) - \tau)(\mathbb{1}(\theta_{i^*} < x) - \mathbb{1}(\theta_{i^*} < y)) = \ell_{\tau,\theta_{i^*}}(x, y),\end{aligned}$$

for $x, y \in \{\frac{j}{m}\}_{j=0}^m$ and $\theta \in [0, 1)$ and $\ell_{\tau,1}(x, y) = 0$. Putting this together with characterization of $\ell \in \mathcal{L}_1$ in Section 4.2.2, we obtain

$$\begin{aligned}& \sup_{\ell \in \mathcal{L}_1, f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^T [\ell(q(t), Y_t) - \ell(f(t), Y_t)] \\ &= \sup_{H_n: \text{pos. measure}, f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^T \sum_{n=1}^N \left[\int_0^1 \ell_{\tau_n, \theta}(q_n(t), Y_t) - \ell_{\tau_n, \theta}(f^n(t), Y_t) dH_n(\theta) \right] \\ &= \sup_{\ell_{\tau,1} \equiv 0} \sup_{H_n, f} \frac{1}{T} \sum_{t=1}^T \sum_{n=1}^N \sum_{i=1}^m [\ell_{\tau_n, \theta_i}(q_n(t), Y_t) - \ell_{\tau_n, \theta_i}(f^n(t), Y_t)] H_n \left(\theta \in \left[\theta_i - \frac{1}{2m}, \theta_i + \frac{1}{2m} \right) \right) \\ &= \sup_{H_n, f} \sum_{n=1}^N \sum_{i=1}^m \frac{1}{T} \sum_{t=1}^T [\ell_{\tau_n, \theta_i}(q_n(t), Y_t) - \ell_{\tau_n, \theta_i}(f^n(t), Y_t)] H_n \left(\theta \in \left[\theta_i - \frac{1}{2m}, \theta_i + \frac{1}{2m} \right) \right) \\ &\leq \sup_{i \in [m], n \in [N], f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^T [\ell_{\tau_n, \theta_i}(q_n(t), Y_t) - \ell_{\tau_n, \theta_i}(f^n(t), Y_t)] \left(\sum_{n=1}^N H_n([0, 1)) \right)\end{aligned}$$

Now we compute an upper bound of $\left(\sum_{n=1}^N H_n([0, 1])\right)$ in terms of maximum possible value of $\ell((x_n)_{n=1}^N, y)$. Observe that

$$\begin{aligned}\ell(0, \dots, 0, 1) &= \sum_{n=1}^N \ell_n(0, 1) = \sum_{n=1}^N \tau_n(g_n(1) - g_n(0)) \stackrel{\text{Thm. 4.1}}{=} \sum_{n=1}^N \tau_n H_n([0, 1]) \\ \ell(1, \dots, 1, 0) &= \sum_{n=1}^N \ell_n(1, 0) = \sum_{n=1}^N (1 - \tau_n)(g_n(1) - g_n(0)) \stackrel{\text{Thm. 4.1}}{=} \sum_{n=1}^N (1 - \tau_n) H_n([0, 1]).\end{aligned}$$

Note that $g_n(1) - g_n(0) = H_n([0, 1])$ from left-continuity of g_n . Since $\max \ell$ is larger than $\ell(0, \dots, 0, 1)$ and $\ell(1, \dots, 1, 0)$, it's also larger than the average of these two, which is $(1/2) \sum_{n=1}^N H_n([0, 1])$. Putting together with the condition $|\ell| \leq 1$, we have $\sum_{n=1}^N H_n([0, 1]) \leq 2$ and this proves

$$\frac{\sup_{\ell \in \mathcal{L}_1, f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^T [\ell(q(t), Y_t) - \ell(f(t), Y_t)]}{\sup_{i, n, f} \frac{1}{T} \sum_{t=1}^T [\ell_{\tau_n, \theta_i}(q_n(t), Y_t) - \ell_{\tau_n, \theta_i}(f^n(t), Y_t)]} \leq 2.$$

It remains to prove that the ratio above is at least 1. It follows from the fact that $\ell_{\tau_n, \theta_i} \in \mathcal{L}_1$. $\square$

C.1.2 Proof of Theorem 4.4

In this proof, we denote $\hat{P}(t) = (\hat{P}_n(t))_{n=1}^N$ by $(P_n^*)_{n=1}^N$. We introduce one lemma and two propositions that are essential for proving that the minimax solution P_n^* satisfies the monotonicity constraint (Property 2 in the theorem statement). First, Lemma C.1 defines a quantity V_n^* , on which our closed-form solution P_n^* depends, for each $n \in [N - 1]$, and shows that there exists a non-decreasing sequence $\{j_n^*\}_{n=1}^{N-1}$, where each j_n^* is an index that minimizes $V_n(j)$.

Lemma C.1. Denote $[m]_0 := [m] \cup \{0\}$ and define a function $V_n(j)$ and a quantity V_n^* for $n \in [N - 1]$ as

$$\begin{aligned}V_n(j) &:= \sum_{i=j+1}^m \sum_{s=1}^n \sum_{b=1}^B w_{ib}^s \mathbb{1}\{\theta_i < f_b^s(t)\} + \sum_{i=1}^j \sum_{s=n+1}^N \sum_{b=1}^B w_{ib}^s (1 - \mathbb{1}\{\theta_i < f_b^s(t)\}) \geq 0, \\ V_n^* &:= \min_{j \in [m]_0} V_n(j).\end{aligned}\tag{C.1}$$

and $V_0^* = V_N^* := 0$. Let J_n^* be the set of indices $j \in [m]_0$ that minimizes the $V_n(j)$.

Then, there exists a sorted sequence $0 = j_0^* \leq j_1^* \leq \dots \leq j_{N-1}^* \leq j_N^* = m$ such that $j_n^* \in J_n^*$ for $n \in [N - 1]$.

Proof of Lemma C.1. Define $w_i^s := (w_{ib}^s)_{b=1}^B$, $\mathbb{1}_i^s := (\mathbb{1}\{\theta_i < f_b^s(t)\})_{b=1}^B$, and $\mathbb{1}_B := (1, \dots, 1) \in \mathbb{R}^B$. Then, V_n^* can be rewritten as

$$V_n^* = \min_{j \in [m]_0} \left(\sum_{i=j+1}^m \sum_{s=1}^n w_i^s \cdot \mathbb{1}_i^s + \sum_{i=1}^j \sum_{s=n+1}^N w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s) \right).$$

Suppose there exists a pair of indices $n, n+1$ such that $j_n^* > j_{n+1}^*$. Using the definition of V_n^* , we have

$$\begin{aligned} j_n^* &= \arg \min_{j \in [m]_0} \left(\sum_{i=j+1}^m \sum_{s=1}^n w_i^s \cdot \mathbb{1}_i^s + \sum_{i=1}^j \sum_{s=n+2}^N w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s) + \sum_{i=1}^j w_i^{n+1} \cdot (\mathbb{1}_B - \mathbb{1}_i^{n+1}) \right), \\ j_{n+1}^* &= \arg \min_{j \in [m]_0} \left(\sum_{i=j+1}^m \sum_{s=1}^n w_i^s \cdot \mathbb{1}_i^s + \sum_{i=1}^j \sum_{s=n+2}^N w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s) + \sum_{i=j+1}^m w_i^{n+1} \cdot \mathbb{1}_i^{n+1} \right). \end{aligned}$$

Two expressions share the first two terms and differ in the last term. We abbreviate these terms as following:

$$\begin{aligned} C_j &:= \sum_{i=j+1}^m \sum_{s=1}^n w_i^s \cdot \mathbb{1}_i^s + \sum_{i=1}^j \sum_{s=n+2}^N w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s) \\ D_j &:= \sum_{i=1}^j w_i^{n+1} \cdot (\mathbb{1}_B - \mathbb{1}_i^{n+1}) \\ E_j &:= \sum_{i=j+1}^m w_i^{n+1} \cdot \mathbb{1}_i^{n+1}. \end{aligned}$$

Note that D_j is non-decreasing in j and E_j is non-increasing in j . Since j_n^* and j_{n+1}^* are the indices that minimize the $V_n(j)$ and $V_{n+1}(j)$, respectively, we have $C_{j_n^*} + D_{j_n^*} \leq C_{j_{n+1}^*} + D_{j_{n+1}^*}$ and $C_{j_n^*} + E_{j_n^*} \geq C_{j_{n+1}^*} + E_{j_{n+1}^*}$. This implies that

$$E_{j_{n+1}^*} - E_{j_n^*} \leq C_{j_n^*} - C_{j_{n+1}^*} \leq D_{j_{n+1}^*} - D_{j_n^*}.$$

Because of the monotonicity of D_j and E_j and the assumption that $j_n^* > j_{n+1}^*$, the leftmost term should be non-negative and the rightmost term should be non-positive. Therefore, we obtain

$$C_{j_n^*} - C_{j_{n+1}^*} = D_{j_n^*} - D_{j_{n+1}^*} = E_{j_n^*} - E_{j_{n+1}^*} = 0 \quad \Rightarrow \quad j_{n+1}^* \in J_n^*, \quad j_n^* \in J_{n+1}^*.$$

By choosing j_{n+1}^* as a minimizer of $V_n(j)$ and j_n^* as a minimizer of $V_{n+1}(j)$, we resolve the unsorted pair of indices. Repeating this process yields a sorted sequence $0 = j_0^* \leq j_1^* \leq \dots \leq j_{N-1}^* \leq j_N^* = m$ and this proves the existence of sorted sequence.

Due to its existence, we can simply obtain a sorted sequence by choosing $j_n^* = \min J_n^*$ for all $n \in [N]$. $\square$

Next, we introduce two propositions. The first proposition (Proposition C.1) constructs $(P_n^*)_{n=1}^N$ and prove their properties. The second proposition (Proposition C.2) then show this $(P_n^*)_{n=1}^N$ is indeed a solution for the minimax program (4.9) in Algorithm 1. Note that Property 1 in C.1 statement below proves Property 1 in the original theorem (Theorem 4.4) statement.

Proposition C.1. *Given a sorted sequence $0 = j_0^* \leq j_1^* \leq \dots \leq j_{N-1}^* \leq j_N^* = m$ from Lemma C.1, there exists a distribution P_n^* for $n \in [N]$ of the form*

$$P_n^* = (1 - C_n)\delta_{k_n/m} + C_n\delta_{(k_n+1)/m}, \quad (\text{C.2})$$

such that the following properties hold:

1. $j_{n-1}^* \leq k_n$
2. $k_n < j_n^*$ if $j_{n-1}^* \neq j_n^*$ and $k_n = j_n^*$ with $C_n = 0$ if $j_{n-1}^* = j_n^*$.
3. The support of P_n^* is contained in $[j_{n-1}^*/m, j_n^*/m]$.
4. $\mathbb{E}_{q_n \sim P_n^*} \left[\sum_{i=1}^m \sum_{b=1}^B w_{ib}^n (\mathbb{1}\{\theta_i < q_n\} - \mathbb{1}\{\theta_i < f_b^n(t)\}) \right] = V_{n-1}^* - V_n^*.$

Proof of Proposition C.1. We use the notation w_i^s , $\mathbb{1}_i^s$, and $\mathbb{1}_B$ from the proof of Lemma C.1. Note that the quantity V_n^* defined in Lemma C.1 is equal to $V_n(j_n^*)$ for all $n \in [N-1]$ and that $V_n(j) \geq V_n(j_n^*)$ for all $j \in [m]_0$.

Using the fact that $\mathbb{1}\{\theta_i < k/m\} = \mathbb{1}\{i \leq k\}$, the last property in the proposition statement is equivalent to

$$\begin{aligned} & V_{n-1}^* - V_n^* \\ &= \mathbb{E}_{q_n \sim P_n^*} \left[\sum_{i=1}^m \sum_{b=1}^B w_{ib}^n (\mathbb{1}\{\theta_i < q_n\} - \mathbb{1}\{\theta_i < f_b^n(t)\}) \right] \\ &= (1 - C_n) \sum_{i=1}^m \sum_{b=1}^B w_{ib}^n (\mathbb{1}\{i \leq k_n\} - \mathbb{1}\{\theta_i < f_b^n(t)\}) \\ &\quad + C_n \sum_{i=1}^m \sum_{b=1}^B w_{ib}^n (\mathbb{1}\{i \leq k_n + 1\} - \mathbb{1}\{\theta_i < f_b^n(t)\}) \\ &= (1 - C_n) \sum_{i=1}^m w_i^n \cdot (\mathbb{1}\{i \leq k_n\} \mathbb{1}_B - \mathbb{1}_i^n) + C_n \sum_{i=1}^m w_i^n \cdot (\mathbb{1}\{i \leq k_n + 1\} \mathbb{1}_B - \mathbb{1}_i^n) \\ &= \sum_{i=1}^{k_n} w_i^n \cdot \mathbb{1}_B + C_n w_{k_n}^n \cdot \mathbb{1}_B - \sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n. \end{aligned}$$

The equality between the first and the last line is equivalent to

$$\sum_{i=1}^{k_n} w_i^n \cdot \mathbb{1}_B + C_n w_{k_n}^n \cdot \mathbb{1}_B = \sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n + V_{n-1}^* - V_n^* = \sum_{i=1}^{j_{n-1}^*} w_i^n \cdot \mathbb{1}_B + V_n(j_{n-1}^*) - V_n(j_n^*), \quad (\text{C.3})$$

where we used the identity $\sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n + V_{n-1}(j) = \sum_{i=1}^j w_i^n \cdot \mathbb{1}_B + V_n(j)$ evaluated at $j = j_{n-1}^*$.

Now we divide the cases into three cases. First, consider the case where $j_{n-1}^* = j_n^*$. In this case, simply $k_n = j_{n-1}^*$ with $C_n = 0$ satisfies the equation (C.3). Therefore, $P_n^* = \delta_{j_{n-1}^*/m}$ satisfies the conditions. Second, consider the case where $j_{n-1}^* < j_n^*$ but $V_n(j_{n-1}^*) = V_n(j_n^*)$. $k_n = j_{n-1}^*$ with $C_n = 0$ again satisfies the equation (C.3) and $k_n = j_{n-1}^* < j_n^*$ in this case. Lastly, consider the case where $V_n(j_{n-1}^*) > V_n(j_n^*)$. We have

$$\begin{aligned} \sum_{i=1}^{j_{n-1}^*} w_i^n \cdot \mathbb{1}_F &< \sum_{i=1}^{j_{n-1}^*} w_i^n \cdot \mathbb{1}_F + V_n(j_{n-1}^*) - V_n(j_n^*) \\ &= \sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n + V_{n-1}^* - V_n^* \\ &= \sum_{i=1}^{j_n^*} w_i^n \cdot \mathbb{1}_F + V_{n-1}(j_{n-1}^*) - V_{n-1}(j_n^*) \leq \sum_{i=1}^{j_n^*} w_i^n \cdot \mathbb{1}_F, \end{aligned}$$

where two equalities used the identity $\sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n + V_{n-1}(j) = \sum_{i=1}^j w_i^n \cdot \mathbb{1}_B + V_n(j)$ evaluated at $j = j_n^* - 1$ and $j = j_n^*$ respectively. This implies that there exists a $j_{n-1}^* \leq k_n < j_n^*$ such that

$$\sum_{i=1}^{k_n} w_i^n \cdot \mathbb{1}_B < \sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n - V_n^* + V_{n-1}^* \quad \text{and} \quad \sum_{i=1}^{k_n+1} w_i^n \cdot \mathbb{1}_B \geq \sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n - V_n^* + V_{n-1}^*. \quad (\text{C.4})$$

With such k_n , we define P_n^* with $C_n = \frac{1}{w_{k_n+1}^n \cdot \mathbb{1}_B} \left(\sum_{i=1}^m w_i^n \cdot \mathbb{1}_i^n - V_n^* + V_{n-1}^* - \sum_{i=1}^{k_n} w_i^n \cdot \mathbb{1}_B \right)$ and this satisfies $\text{supp}(P_n^*) \in [j_{n-1}^*/m, j_n^*/m]$ since $k_n + 1 \leq j_n^*$. $\square$

The following theorem shows that P_n^* is a solution for the minimax program (4.9) in Algorithm 1.

Proposition C.2. *Given quantile levels $\tau_1 < \tau_2 < \dots < \tau_N$ and quantile forecasts $\hat{f}_b^n(t)$ for $b \in [B]$, $n \in [N]$, define V_n^* for $n \in [N-1]$ as in equation (C.1), with $V_0^* = V_N^* := 0$. Due to Lemma C.1, there exists an increasing sequence $\{j_n^*\}_{n=1}^{N-1}$ where j_n^* is an index that minimizes*

$V_n(j)$. Define $j_0^* = 0$ and $j_N^* = m$.

Then, P_n^* for $n \in [N]$ given by (C.2) with

$$P_y^* = \tau_1 \delta_0 + (1 - \tau_N) \delta_1 + \sum_{n=1}^{N-1} (\tau_{n+1} - \tau_n) \delta_{j_n^*/m}$$

is a solution of the minimax program (4.9) and the value of the minimax program is

$$-\sum_{n=1}^{N-1} (\tau_{n+1} - \tau_n) V_n^* \leq 0.$$

Proof of Proposition C.2. Denote the minimax objective function by

$$O((P_n)_{n=1}^N, P_y) = \sum_{n=1}^N \sum_{i=1}^m \sum_{b=1}^B w_{ib}^n \mathbb{E}_{q_n \sim P_n, Y \sim P_y} [\ell_{\tau_n, \theta_i}(q_n, Y) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y)].$$

It suffices to prove $O((P_n^*)_{n=1}^N, P_y^*) = \max_{P_y} O((P_n^*)_{n=1}^N, P_y)$ then prove $O((P_n^*)_{n=1}^N, P_y^*) = \min_{P_n \in \Delta_m, n \in [N]} O((P_n)_{n=1}^N, P_y^*)$.

We start with decomposing $O((P_n)_{n=1}^N, P_y)$. First, observe the following equality:

$$\begin{aligned} & \ell_{\tau_n, \theta_i}(q_n, Y) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y) \\ &= (\mathbb{1}(Y < q_n) - \tau_n)(\mathbb{1}(\theta_i < q_n) - \mathbb{1}(\theta_i < Y)) \\ & \quad - (\mathbb{1}(Y < f_b^n(t)) - \tau_n)(\mathbb{1}(\theta_i < f_b^n(t)) - \mathbb{1}(\theta_i < Y)) \\ &= \mathbb{1}(Y, \theta_i < q_n) - \tau_n \mathbb{1}(\theta_i < q_n) - \mathbb{1}(\theta_i < Y < q_n) \\ & \quad - \mathbb{1}(Y, \theta_i < f_b^n(t)) + \tau_n \mathbb{1}(\theta_i < f_b^n(t)) + \mathbb{1}(\theta_i < Y < f_b^n(t)) \\ &= \mathbb{1}(Y \leq \theta_i < p) - \tau_n \mathbb{1}(\theta_i < q_n) - \mathbb{1}(Y \leq \theta_i < f_b^n(t)) + \tau_n \mathbb{1}(\theta_i < f_b^n(t)). \end{aligned} \quad (\text{C.5})$$

Then, $O((P_n)_{n=1}^N, P_y)$ can be written as

$$\begin{aligned} & O((P_n)_{n=1}^N, P_y) \\ &= \sum_{n=1}^N \sum_{i=1}^m \sum_{b=1}^B w_{ib}^n \mathbb{E}_{q_n \sim P_n, Y \sim P_y} [\mathbb{1}(Y \leq \theta_i < q_n) - \tau_n \mathbb{1}(\theta_i < q_n) - \mathbb{1}(Y \leq \theta_i < f_b^n(t))] + C_n, \end{aligned} \quad (\text{C.6})$$

where $C_n = \tau_n \sum_{i=1}^m \sum_{b=1}^B w_{ib}^n \mathbb{1}\{\theta_i < f_b^n(t)\}$ which does not depend on P_n or P_y .

Using this decomposition, we prove $O((P_n^*)_{n=1}^N, P_y^*) = \max_{P_y} O((P_n^*)_{n=1}^N, P_y)$. Let $P_y(\frac{j}{m}) = p_{yj}$ for $j = 0, \dots, m$. Then, again using the notation w_i^s , $\mathbb{1}_i^s$, and $\mathbb{1}_B$ from the proof of Lemma C.1, $O((P_n^*)_{n=1}^N, P_y)$ is rephrased as the following:

$$O((P_n^*)_{n=1}^N, P_y) = \sum_{n=1}^N \mathbb{E}_{q_n \sim P_n^*} \left[\sum_j p_{yj} \sum_{i=j+1}^m w_i^n \cdot (\mathbb{1}\{\theta_i < q_n\} \mathbb{1}_F - \mathbb{1}_i^n) \right] + C(P_n^*), \quad (\text{C.7})$$

where $C(P_n^*) = -\tau_n \mathbb{E}_{q_n \sim P_n^*} \sum_{i=1}^m w_i^n \cdot (\mathbb{1}\{\theta_i < q_n\} \mathbb{1}_F - \mathbb{1}_i^n)$ only depends on P_n^* and does not depend on P_y .

Plugging in P_n^* given as (C.2), the summand can be expressed as

$$\begin{aligned} A_j^n &:= \mathbb{E}_{q_n \sim P_n^*} \left[\sum_{i=j+1}^m w_i^n \cdot (\mathbb{1}\{\theta_i < q_n\} \mathbb{1}_F - \mathbb{1}_i^n) \right] \\ &= \begin{cases} -\sum_{i=1}^j w_i^n \cdot (\mathbb{1}_B - \mathbb{1}_i^n) - V_n^* + V_{n-1}^* & j \leq k_n^* \\ -\sum_{i=j+1}^m w_i^n \cdot \mathbb{1}_i^n & j > k_n^* \end{cases} \end{aligned}$$

This leads to

$$\begin{aligned} A_j &:= \sum_{n=1}^N A_j^n \\ &= \begin{cases} -\sum_{n=1}^N \sum_{i=1}^j w_i^n \cdot (\mathbb{1}_B - \mathbb{1}_i^n) + V_0^* & j \leq k_1 \\ -\sum_{s=1}^n \sum_{i=j+1}^m w_i^s \cdot \mathbb{1}_i^s \\ \quad - \sum_{s=n+1}^N \sum_{i=1}^j w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s) + V_n^* & j \in (k_n, k_{n+1}], n \in [N-1] \\ -\sum_{n=1}^N \sum_{i=j+1}^m w_i^n \cdot \mathbb{1}_i^n + V_N^* & j > k_N \end{cases} \end{aligned}$$

Notice that A_j is non-increasing on $j \leq k_1$ and non-decreasing on $j > k_N$. $A_0 = 0$ and $A_m = 0$ is the maximum value of A_j in these ranges, respectively. Next, observe that $A_j \leq 0$ for all $j = 0, 1, \dots, m$ since V_n^* is the minimum value that $\sum_{s=1}^n \sum_{i=j+1}^m w_i^s \cdot \mathbb{1}_i^s + \sum_{s=n+1}^N \sum_{i=1}^j w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s)$ can achieve by its definition. It suffices to show that $A_{j_n^*} = 0$ for $n \in [N-1]$ since P_y^* only puts mass on these points. Note that $A_{j_0} = A_0 = 0$ and $A_{j_N} = A_m = 0$.

From the first and second properties in the statement of Proposition C.1, we have $k_n < j_n^* \leq k_{n+1}$ if $j_{n-1}^* \neq j_n^*$ and $j_{n-1}^* = j_n^* = k_n$ if $j_{n-1}^* = j_n^*$, for $n \in [N-1]$. In the former case, we obtain $A_{j_n^*} = 0$ from $j \in (k_n, k_{n+1}]$ case in the definition of A_j above. In the latter case, $j_n^* = k_n$ does not fall into the range $(k_n, k_{n+1}]$. Still, from $j \in (k_{n-1}, k_n]$ case ($j \leq k_1$ case if $n = 1$) in the definition of A_j , we have

$$\begin{aligned} A_{j_n^*} &= -\sum_{s=1}^{n-1} \sum_{i=j+1}^m w_i^s \cdot \mathbb{1}_i^s - \sum_{s=n}^N \sum_{i=1}^j w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s) + V_{n-1}^* \\ &= -V_{n-1}(j_n^*) + V_{n-1}^* = -V_{n-1}(j_{n-1}^*) + V_{n-1}^* = 0, \end{aligned}$$

where $V_n(j)$ is defined as equation (C.1). Therefore, $A_{j_n^*} = 0$ for all $n \in [N-1]$ and $O((P_n^*)_{n=1}^N, P_y^*) = \max_{P_y} O((P_n^*)_{n=1}^N, P_y)$.

Next, we show that $O((P_n^*)_{n=1}^N, P_y^*) = \min_{P_n \in \Delta_m, n \in [N]} O((P_n)_{n=1}^N, P_y^*)$. This follows directly from the propriety of ℓ_{τ_n, θ_i} losses and the third property in Proposition C.1. Any point in the support of P_n^* is τ_n -level quantile of P_y^* since $P_y^*(Y \leq j_{n-1}^*/m) = \tau_n$ and $P_y^*(Y \geq j_n^*/m) = \tau_n$.

Lastly, we compute the value of the minimax program. Decomposition (C.7) can be rewritten as

$$O((P_n^*)_{n=1}^N, P_y) = \sum_{j=0}^m p_{yj} A_j + \sum_{n=1}^N C(P_n^*).$$

Since P_y^* only puts mass on points j/m such that $A_j = 0$, plugging in P_y^* into the equation above yields

$$\begin{aligned} O((P_n^*)_{n=1}^N, P_y^*) &= \sum_{n=1}^N C(P_n^*) = - \sum_{n=1}^N \tau_n \mathbb{E}_{q_n \sim P_n^*} \sum_{i=1}^m w_i^n \cdot (\mathbb{1}\{\theta_i < q_n\} \mathbb{1}_F - \mathbb{1}_i^n) \\ &= - \sum_{n=1}^N \tau_n (V_{n-1}^* - V_n^*) = - \sum_{n=1}^{N-1} (\tau_{n+1} - \tau_n) V_n^* \leq 0, \end{aligned}$$

where the second from the last equality used the fourth property in Proposition C.1. $\square$

Lastly, we prove the second and third property in Theorem 4.4 statment using Proposition C.1 and Proposition C.2.

Proof of Property 2 in Theorem 4.4. It suffices to prove that P_{n+1} is stochastically larger than P_n since we can couple these two by $U \sim \text{Unif}(0, 1)$ and $P_n = F_{P_n}^{-1}(U)$ and $P_{n+1} = F_{P_{n+1}}^{-1}(U)$ where F_X^{-1} denotes a inverse of cumulative distribution functon of X .

Combining the first three properties in the statement in Proposition C.1, we have $k_n < j_n^* \leq k_{n+1}$ if $j_{n-1}^* \neq j_n^*$. In this case, P_{n+1}^* is stochastically larger than P_n^* since P_n^* is supported on $\{k_n/m, (k_n + 1)/m\}$ and $k_{n+1} > k_n$. In case $j_{n-1}^* = j_n^*$, $P_n^* \equiv j_n^*$ and P_{n+1} is stochastically larger than P_n since $k_{n+1} \geq j_n^*$. $\square$

Proof of Property 3 in Theorem 4.4. Since P_n^* is stochastically ordered, it suffices to prove that $P_1^* \leq m_1/m$ and $P_N^* \geq m_2/m$ almost surely. First we prove that $P_1^* \geq m_1/m$ almost surely. Observe that, for $i \leq m_1$, $\mathbb{1}\{\theta_i < f_b^n(t)\} = 1$ for all $b \in [B]$ and $n \in [N]$. This implies that $\mathbb{1}_i^n = \mathbb{1}_B$ for all $n \in [N]$ and $i \leq m_1$. Therefore, for $j \leq m_1$,

$$V_1(j) = \sum_{i=j+1}^m w_i^1 \cdot \mathbb{1}_i^1 + \sum_{i=1}^j \sum_{s=2}^N w_i^s \cdot (\mathbb{1}_B - \mathbb{1}_i^s) = \sum_{i=j+1}^{m_1} w_i^1 \cdot \mathbb{1}_B + \sum_{i=m_1+1}^m w_i^1 \cdot \mathbb{1}_i^1$$

holds and $V_1(j)$ is non-increasing in j on $j \leq m_1$. This implies that $V_1^* \leq V_1(m_1) = \sum_{i=m_1+1}^m w_i^1 \cdot \mathbb{1}_i^1$ and thus

$$\sum_{i=1}^{m_1} w_i^1 \cdot \mathbb{1}_B = \sum_{i=1}^{m_1} w_i^1 \cdot \mathbb{1}_i^1 \leq \sum_{i=1}^m w_i^1 \cdot \mathbb{1}_i^1 - V_1^* + V_0^*,$$

since $\mathbb{1}_i^s = \mathbb{1}_B$ for $i \leq m_1$ and $V_0^* = 0$. In case the inequality is strict, k_1 that satisfies the condition (C.4) should be at least m_1 , which proves $P_1^* \geq m_1/m$ almost surely. In case the equality holds, $k_1 + 1 = m_1$ since $w_{m_1}^1 \cdot \mathbb{1}_B = \sum_{b=1}^B w_{m_1}^1 > 0$ from the corollary condition. However, $C_1 = 1$ in this case and $P_1^* = \delta_{m_1/m}$ which satisfies $P_1^* \geq m_1/m$ almost surely.

Next, we prove that $P_N^* \leq m_2/m$ almost surely in a similar manner. For $i > m_2$, $\mathbb{1}\{\theta_i < f_b^n(t)\} = 0$ for all $b \in [B]$ and $n \in [N]$. This implies that $\mathbb{1}_i^n = \mathbb{1}_B$ for all $n \in [N]$ and $i > m_2$. Therefore, for $j \geq m_2$,

$$V_{N_1}(j) = \sum_{i=j+1}^m \sum_{s=1}^{N-1} w_i^s \cdot \mathbb{1}_i^s + \sum_{i=1}^j w_i^N \cdot (\mathbb{1}_B - \mathbb{1}_i^N) = \sum_{i=1}^{m_2} w_i^N \cdot (\mathbb{1}_B - \mathbb{1}_i^N) + \sum_{i=m_2+1}^m w_i^N \cdot \mathbb{1}_B$$

holds and $V_{N-1}(j)$ is non-decreasing in j on $j \geq m_2$. This implies that $V_{N-1} \leq V_{N-1}(m_2) = \sum_{i=1}^{m_2} w_i^N \cdot (\mathbb{1}_B - \mathbb{1}_i^N)$ and thus

$$\sum_{i=1}^m w_i^N \cdot \mathbb{1}_i^N - V_N^* + V_{N-1}^* \leq \sum_{i=1}^m w_i^N \cdot \mathbb{1}_i^N + \sum_{i=1}^{m_2} w_i^N \cdot (\mathbb{1}_B - \mathbb{1}_i^N) = \sum_{i=1}^{m_2} w_i^N \cdot \mathbb{1}_B$$

since $\mathbb{1}_i^N = \mathbf{0}$ for $i > m_2$ and $V_N^* = 0$. Because $w_{m_2}^N \cdot \mathbb{1}_B = \sum_{b=1}^B w_{m_2}^N > 0$ from the corollary condition, k_n that satisfies the condition (C.4) should satisfy $k_N < m_2$, which proves $P_N^* \leq m_2/m$ almost surely. $\square$

C.1.3 Proof of Theorem 4.3

Theorem C.1 (Full statement of Theorem 4.3). *For a set of quantile levels $\tau_1 < \tau_2 < \dots < \tau_N$, let $\mathcal{L}_1$ denote the class of proper losses for multiple quantile levels defined as equation (4.6). Consider base forecasters $\{f_b\}_{b=1}^B$, where each forecaster f_b provides $\{\tau_n\}_{n=1}^N$ -level quantile forecasts $\{f_b^n(t)\}_{n=1}^N$ at each time $t \in [T]$. With the target values Y_t , forecaster predictions $f_b^n(t)$ for $t \in [T], n \in [N], b \in [B]$ each taking values in a uniform grid of size m , and the learning rate $\eta = c\sqrt{(\log mNB)/T}$, the output $\hat{P}(t) = (\hat{P}_n(t))_{n=1}^N$ from Algorithm 1 achieves the following omniprediction error:*

$$\sup_{\ell \in \mathcal{L}_1, b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q_n \sim \hat{P}_n(t), \forall n} [\ell((q_n)_{n=1}^N, Y_t)] - \ell(f_b(t), Y_t) \leq 2 \left(c + \frac{1}{c} \right) \left(\sqrt{\frac{\log mNB}{T}} \right)$$

Proof. Observe that $\ell_{\tau_n, \theta_i}(x, y) \in \{0, \tau_n, 1 - \tau_n\}$ and thus $|\ell_{\tau_n, \theta_i}(x_1, y_1) - \ell_{\tau_n, \theta_i}(x_2, y_2)| \leq 1$. Without loss of generality, we assume that $Y_t, f_b(t) \in \{0, \frac{1}{m}, \dots, 1\}$ for all $t \in [T], b \in [B]$, as this construction extends to any bounded interval $[-B_\ell, B_u]$ via rescaling.

Regret bound of hedge algorithm (Theorem C.4) gives the following:

$$\begin{aligned}
& \max_{i \in [m], n \in [N], b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q_n \sim \hat{P}_n(t), \forall n} [\ell_{\tau_n, \theta_i}(q_n, Y_t) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)] \\
& \leq \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^m \sum_{n=1}^N \sum_{b=1}^B w_{ib}^n(t) \mathbb{E}_{q_n \sim \hat{P}_n(t), \forall n} [\ell_{\tau_n, \theta_i}(q_n, Y_t) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)] + \eta + \frac{\log mNB}{T\eta}
\end{aligned} \tag{C.8}$$

Proposition C.2 states that the value of the minimax program (4.9) is non-positive. Putting these together with previous lemmas, we have:

$$\begin{aligned}
& \sup_{\ell \in \mathcal{L}_1, f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q_n \sim \hat{P}_n(t), \forall n} [\ell(\{q_n\}_{n=1}^N, Y_t) - \ell(f(t), Y_t)] \\
& \stackrel{\text{Lem. 4.1}}{\leq} 2 \sup_{i \in [m], n \in [N], b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q_n \sim \hat{P}_n(t), \forall n} [\ell_{\tau_n, \theta_i}(q_n, Y_t) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)] \\
& \stackrel{\text{Ieq. (C.8)}}{\leq} \frac{2}{T} \sum_{t=1}^T \sum_{i=1}^m \sum_{n=1}^N \sum_{b=1}^B w_{ib}^n(t) \mathbb{E}_{q_n \sim \hat{P}_n(t), \forall n} [\ell_{\tau_n, \theta_i}(q_n, Y_t) - \ell_{\tau_n, \theta_i}(f_b^n(t), Y_t)] \\
& \quad + 2 \left(c + \frac{1}{c} \right) \left(\sqrt{\frac{\log mNB}{T}} \right) \\
& \stackrel{\text{Prop. C.2}}{\leq} 2 \left(c + \frac{1}{c} \right) \left(\sqrt{\frac{\log mNB}{T}} \right).
\end{aligned}$$

□

C.2 Single quantile omniprediction for proper losses

In this section, we consider a single quantile level, $\tau \in (0, 1)$. We estimate the τ quantile of the ground truth Y_t with $q(t) \in \mathbb{R}$, and evaluate the performance of these predictions by comparing them to a finite set of B base forecasters $\mathcal{F} = \{\{f_b(t)\}_{t \in [T]} : b \in [B]\}$, where $f_b(t) \in [0, 1]$ denotes the forecast of the τ -quantile made by base forecaster f_b at time step t .

We define the following class of loss functions $\ell : [0, 1] \times [0, 1] \rightarrow \mathbb{R}_{\geq 0}$, which is an analogous version of $\mathcal{L}_1$ (defined in (4.6)) for proper losses for single quantile level:

$$\mathcal{L}_\tau := \{ \ell \mid \ell(q, y) = (\mathbb{1}(y < q) - \tau)(g(q) - g(y)), \|\ell\|_\infty \leq 1 \}, \tag{C.9}$$

where g is non-decreasing and left-continuous. As discussed in Section 4.2.2, $\mathcal{L}_\tau$ is **exhaustive** in the sense that any proper loss function for the τ -level quantile that satisfies the mild regularity conditions can be expressed in this form.

C.2.1 Discretization

Similarly to Lemma 4.1, the following lemma shows that to control the omniprediction error over the class $\mathcal{L}_\tau$, it suffices to control the error over the finite collection of elementary losses $\{\ell_{\tau, \theta_i}\}_{i \in [m]}$.

Lemma C.2. *Suppose $Y_t, q(t), f_b(t) \in \{0, 1/m, \dots, 1\}$ for all $t \in [T]$, and $b \in [B]$. Then,*

$$1 \leq \frac{\sup_{\ell \in \mathcal{L}_\tau, b} \frac{1}{T} \sum_{t=1}^T [\ell(q(t), Y_t) - \ell(f_b(t), Y_t)]}{\sup_{i, b} \frac{1}{T} \sum_{t=1}^T [\ell_{\tau, \theta_i}(q(t), Y_t) - \ell_{\tau, \theta_i}(f_b(t), Y_t)]} \leq \frac{1}{\max\{\tau, 1 - \tau\}}.$$

Proof. The proof follows the proof of Lemma 4.1. One additional step is to observe that

$$\begin{aligned} \max_{q, y} \ell(q, y) &= \max_{q, y} (\mathbb{1}(y < q) - \tau)(g(q) - g(y)) \\ &= \max\{\tau, 1 - \tau\}(g(1) - g(0)) = \max\{\tau, 1 - \tau\}H([0, 1]). \end{aligned}$$

This implies $H([0, 1]) = \frac{1}{\max\{\tau, 1 - \tau\}} \max_{q, y} \ell(q, y) \leq \frac{1}{\max\{\tau, 1 - \tau\}}$, which concludes the proof. $\square$

C.2.2 Omniprediction algorithm and theoretical guarantee

This subsection introduces the omniprediction algorithm that optimizes over all $\ell \in \mathcal{L}_\tau$. We then prove a theoretical upper bound for the omniprediction error of the algorithm.

Algorithm 2 Single quantile level omniprediction

- 1: **Data:** Data stream $(Y_t)_{t=1}^T$, Grid size $m \in \mathbb{N}$, Learning rate $\eta > 0$, Base forecasters $\mathcal{F} = \{f_b\}_{b=1}^B$.
 - 2: For $t = 1, w_{ib}(t) = \frac{1}{mB}$, for all $i \in [m], b \in [B]$;
 - 3: **for** $t = 1, \dots, T$ **do**
 - 4: **1) Make prediction at time t :**
 - 5: $B_k := \sum_{i=1}^k \sum_{b=1}^B w_{ib}(t) - \sum_{i=1}^m \sum_{b=1}^B w_{ib}(t) \mathbb{1}(\theta_i < f_b(t))$, $k^* = \max\{k : B_k \leq 0\}$
 - 6: **Output** $\hat{P}(t) = \frac{B_{k^*+1}}{B_{k^*+1} + |B_{k^*}|} \delta_{\frac{k^*}{m}} + \frac{|B_{k^*}|}{B_{k^*+1} + |B_{k^*}|} \delta_{\frac{k^*+1}{m}}$
 - 7:
 - 8: **2) Update the weights for each ℓ_{θ_i} and f_r :**
 - 9: $w_{ib}(t+1) = \frac{w_{ib}(t) \exp\left(\eta \left(\mathbb{E}_{q \sim \hat{P}(t)} [\ell_{\tau, \theta_i}(q, Y_t)] - \ell_{\tau, \theta_i}(f_b(t), Y_t)\right)\right)}{\sum_{i, b} w_{ib}(t) \exp\left(\eta \left(\mathbb{E}_{q \sim \hat{P}(t)} [\ell_{\tau, \theta_i}(q, Y_t)] - \ell_{\tau, \theta_i}(f_b(t), Y_t)\right)\right)}$, $\forall i \in [m], b \in [B]$;
 - 10:
 - 11: **end for**
-

The first part of the algorithm is to produce a prediction at time t . Letting Δ_m denote the set of probability distributions over $\{0, \frac{1}{m}, \frac{2}{m}, \dots, 1\}$, the prediction $\hat{P}(t)$ computes a minimax solution to the following program:

$$\hat{P}(t) = \arg \min_{P \in \Delta_m} \max_{P_y \in \Delta_m} \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbb{E}_{q \sim P, Y \sim P_y} [\ell_{\tau, \theta_i}(q, Y) - \ell_{\tau, \theta_i}(f_b(t), Y)] \quad (\text{C.10})$$

The following proposition provides a closed-form solution to the minimax program (C.10), which is incorporated in Algorithm 2, by exploiting the structure of ℓ_{τ, θ_i} losses.

Proposition C.3. Define $B_k := \sum_{i=1}^k \sum_{b=1}^B w_{ib} - \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbf{1}(\theta_i < f_b(t))$ and let $k^* = \max\{k : B_k \leq 0\}$. Note that $B_{k^*} \leq 0$ and $B_{k^*+1} > 0$. Then,

$$P^* = \frac{B_{k^*+1}}{B_{k^*+1} + |B_{k^*}|} \delta_{\frac{k^*}{m}} + \frac{|B_{k^*}|}{B_{k^*+1} + |B_{k^*}|} \delta_{\frac{k^*+1}{m}} \quad \text{and} \quad P_y^* = \tau \delta_0 + (1 - \tau) \delta_1$$

solves equation (C.10), and the value of the minimax program is 0.

Proof. Let $O(P, P_y) := \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbb{E}_{q \sim P, Y \sim P_y} [\ell_{\tau, \theta_i}(q, Y) - \ell_{\tau, \theta_i}(f_b(t), Y)]$. It suffices to prove that $O(P^*, P_y^*) = \max_{P_y} O(P^*, P_y) = \min_P O(P, P_y^*)$. Using the decomposition result from equation (C.5), we have

$$O(P, P_y) = \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbb{E}_{q \sim P, Y \sim P_y} \left[\mathbf{1}(Y \leq \theta_i < q) - \tau \mathbf{1}(\theta_i < q) - \mathbf{1}(Y \leq \theta_i < f_b(t)) \right] + C_\tau. \quad (\text{C.11})$$

where $C_\tau = \tau \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbf{1}(\theta_i < f_b(t))$ which does not depend on P or P_y .

We first prove $O(P^*, P_y^*) = \max_{P_y} O(P^*, P_y)$. Let $P_y(\frac{j}{m}) = p_{yj}$ for $j = 0, \dots, m$. Using the fact that $\mathbf{1}(\theta_i < k/m) = \mathbf{1}(i \leq k)$, $O(P^*, P_y)$ is rephrased as the following:

$$\begin{aligned} O(P^*, P_y) &= \mathbb{E}_{q \sim P^*} \left[\sum_{j=0}^m p_{yj} \left(\sum_{i=j+1}^m \sum_{b=1}^B w_{ib} (\mathbf{1}(\theta_i < q) - \mathbf{1}(\theta_i < f_b(t))) \right) \right] + C(P^*) \\ &= \frac{B_{k^*+1}}{B_{k^*+1} + |B_{k^*}|} \left[\sum_{j=0}^m p_{yj} \left(\sum_{i=j+1}^{k^*} \sum_{b=1}^B w_{ib} - \sum_{i=j+1}^m \sum_{b=1}^B w_{ib} \mathbf{1}(\theta_i < f_b(t)) \right) \right] \\ &\quad + \frac{|B_{k^*}|}{B_{k^*+1} + |B_{k^*}|} \left[\sum_{j=0}^m p_{yj} \left(\sum_{i=j+1}^{k^*+1} \sum_{b=1}^B w_{ib} - \sum_{i=j+1}^m \sum_{b=1}^B w_{ib} \mathbf{1}(\theta_i < f_b(t)) \right) \right] + C(P^*), \end{aligned} \quad (\text{C.12})$$

where $C(P^*)$ is a value that does not depend on P_y and we ignore the term hereafter.

Now observe that $A_k^j := \sum_{i=j+1}^k \sum_{b=1}^B w_{ib} - \sum_{i=j+1}^m \sum_{b=1}^B w_{ib} \mathbf{1}(\theta_i < f_b(t))$ is non-increasing on $j \leq k$ and non-decreasing on $j \in [k, m]$ since $\mathbf{1}(\theta_i < f_b(t))$ is 0 or 1. Also note that $A_k^0 \geq A_k^{m-1}$ if $k = m$ and $A_k^m = 0 \geq A_k^{m-1}$ if $k \neq m$. Therefore, A_k^j achieves maximum at

$j = 0$ or $j = m$:

$$\begin{aligned}
& \max_{\sum_j p_{yj}=1} \sum_{j=0}^m p_{yj} \left(\sum_{i=j+1}^k \sum_{b=1}^B w_{ib} - \sum_{i=j+1}^m \sum_{b=1}^B w_{ib} \mathbb{1}(\theta_i < f_b(t)) \right) \\
&= \max_{p_{y0}+p_{ym}=1} p_{y0} \left(\sum_{i=1}^k \sum_{b=1}^B w_{ib} - \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbb{1}(\theta_i < f_b(t)) \right) + p_{ym} \cdot 0 \\
&= \max_{p_{y0}+p_{ym}=1} p_{y0} \cdot B_k + p_{ym} \cdot 0.
\end{aligned}$$

Putting this together with equation (C.12) and the fact that $B_{k^*} \leq 0$ gives

$$\begin{aligned}
& \max_{P_y} O(P^*, P_y) \\
&= \max_{p_{y0}+p_{ym}=1} p_{y0} \left(\frac{B_{k^*+1}}{B_{k^*+1} + |B_{k^*}|} \cdot B_{k^*} + \frac{|B_{k^*}|}{B_{k^*+1} + |B_{k^*}|} \cdot B_{k^*+1} \right) + p_{ym} \cdot 0 + C(P^*),
\end{aligned}$$

which is achieved by P_y^* since P_y^* only puts mass on $j = 0, m$. This proves that $O(P^*, P_y^*) = \max_{P_y} O(P^*, P_y)$.

Next, we prove $O(P^*, P_y^*) = \min_P O(P, P_y^*)$. The decomposition (C.11) of $O(P, P_y)$ reduces to

$$\begin{aligned}
O(P, P_y^*) &= \tau \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbb{E}_{q \sim P} [(1 - \tau) \mathbb{1}(\theta_i < q) - \mathbb{1}(\theta_i < f_b(t))] + \\
&\quad (1 - \tau) \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbb{E}_{q \sim P} [-\tau \mathbb{1}(\theta_i < q)] + C_\tau \\
&= -\tau \sum_{i=1}^m \sum_{b=1}^B w_{ib} \mathbb{1}(\theta_i < f_b(t)) + C = 0,
\end{aligned}$$

which does not depend on P . Thus, $O(P^*, P_y^*) = \min_P O(P, P_y^*)$. $\square$

Now, we present the omniprediction error bound for Algorithm 2.

Theorem C.2. *For a single quantile level τ , let $\mathcal{L}_\tau$ denote the class of proper losses for τ quantile levels defined as equation (C.9). Consider base forecasters $\{f_b\}_{b=1}^B$, where each forecaster f_b provides τ -level quantile forecasts $f_b(t)$ at each time $t \in [T]$. With the target values Y_t , base forecasts $f_b(t)$ for $t \in [T], b \in [B]$ each taking values in a uniform grid of size m , and the learning rate $\eta = c\sqrt{(\log mB)/T}$, the output $\hat{P}(t)$ from Algorithm 2 achieves the following omniprediction error:*

$$\sup_{\ell \in \mathcal{L}_\tau, b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q \sim \hat{P}(t)} [\ell(q, Y_t) - \ell(f_b(t), Y_t)] \leq \frac{1}{\max\{\tau, 1 - \tau\}} \left(c + \frac{1}{c} \right) \sqrt{\frac{\log mB}{T}}$$

Proof. The proof follows the proof in Appendix C.1.3. Regret bound of hedge algorithm (Theorem C.4) gives the following:

$$\begin{aligned} & \max_{i \in [m], b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q \sim \hat{P}(t)} [\ell_{\tau, \theta_i}(q, Y_t) - \ell_{\tau, \theta_i}(f_b(t), Y_t)] \\ & \leq \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^m \sum_{b=1}^B w_{ib}(t) \mathbb{E}_{q \sim \hat{P}(t)} [\ell_{\tau, \theta_i}(q, Y_t) - \ell_{\tau, \theta_i}(f_b(t), Y_t)] + \eta + \frac{\log mB}{T\eta}. \end{aligned} \quad (\text{C.13})$$

Putting this together with previous lemmas, we have

$$\begin{aligned} & \sup_{\ell \in \mathcal{L}_\tau, f \in \mathcal{F}} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q \sim \hat{P}(t)} [\ell(q, Y_t)] - \ell(f(t), Y_t) \\ & \leq \frac{1}{\max\{\tau, 1 - \tau\}} \sup_{i \in [m], b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q \sim \hat{P}(t)} [\ell_{\tau, \theta_i}(q, Y_t)] - \ell_{\tau, \theta_i}(f_b(t), Y_t), \end{aligned}$$

and

$$\begin{aligned} & \sup_{i \in [m], b \in [B]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q \sim \hat{P}(t)} [\ell_{\tau, \theta_i}(q, Y_t)] - \ell_{\tau, \theta_i}(f_b(t), Y_t) \\ & \stackrel{\text{Ieq. (C.13)}}{\leq} \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^m \sum_{b=1}^B w_{ib}(t) \mathbb{E}_{q \sim \hat{P}(t)} [\ell_{\tau, \theta_i}(q, Y_t) - \ell_{\tau, \theta_i}(f_b(t), Y_t)] + \left(c + \frac{1}{c}\right) \left(\sqrt{\frac{\log mB}{T}}\right) \\ & \stackrel{\text{Prop. C.3}}{\leq} \left(c + \frac{1}{c}\right) \left(\sqrt{\frac{\log mB}{T}}\right), \end{aligned}$$

where the last inequality used the fact that the value of the minimax program is 0. This concludes the proof. $\square$

C.3 Quantile omniprediction for weighted quantile losses

In this section, we develop omnipredictors for the restricted subset of proper loss functions that can be expressed as weighted combinations of pinball losses. Although this is a much smaller set compared to the set of all proper losses for multiple quantile levels, which we discussed in Section 4.3, it has the advantage of not requiring any discretization of the target outcome or base forecasters' predictions.

We introduce two algorithms for omniprediction for weighted quantile losses. In Appendix C.3.1, we discuss the first algorithm which runs the Hedge algorithm on each quantile level then combines quantile estimates from each level. In Appendix C.3.2, we introduce a two-player game based algorithm that has a similar form to the omniprediction algorithm for multiple quantile levels given in Algorithm 1. Finally, in Appendix C.3.3, we compare the two algorithm in terms of time complexity, quantile crossing, and omniprediction error bound.

Notation Define the pinball loss at quantile level τ as $\rho_\tau(x) := \tau x \mathbb{1}\{x \geq 0\} + (1-\tau)|x| \mathbb{1}\{x < 0\} = (\tau - \mathbb{1}\{x < 0\})x$. Weighted quantile loss at quantile levels $\{\tau_n\}_{n=1}^N$ is defined as

$$\ell((x_n)_{n=1}^N, y) := \sum_{n=1}^N w_n \rho_{\tau_n}(y - x_n),$$

where w_n is the weight for the τ_n -level quantile prediction and $\sum_{n=1}^N w_n = 1$. We denote $\mathcal{L}_{wql}$ as the set of all weighted quantile loss functions and use $\rho_n(x, y)$ to denote $\rho_{\tau_n}(y - x)$ for $n \in [N]$. We define $f_{\min}(t) = \min_{n \in [N], b \in [B]} f_b^n(t)$ and $f_{\max}(t) = \max_{n \in [N], b \in [B]} f_b^n(t)$ for each time t .

We aim to come up with $q(t) = (q_n(t))_{n=1}^N$ that minimizes the following omniprediction objective:

$$\sup_{\ell \in \mathcal{L}_{wql}, b \in [B]} \frac{1}{T} \sum_{t=1}^T [\ell(q(t), Y_t) - \ell(f_b(t), Y_t)],$$

where $f_b(t) := (f_b^n(t))_{n=1}^N$.

C.3.1 Hedge algorithm on each quantile level

In this subsection, we introduce the first algorithm that essentially runs Hedge algorithm on each quantile level then provide the omniprediction error bound for this algorithm. We denote the output from Algorithm 3 by $q^H(t) := (q_n^H(t))_{n=1}^N$ where the superscript H denotes it is from the Hedge algorithm.

Algorithm 3 Hedge algorithm for multiple quantile levels in parallel

- 1: **Data:** Data stream $(Y_t)_{t=1}^T$, Quantile levels $\{\tau_n\}_{n=1}^N$, Learning rate $\eta > 0$, Base forecasters: $\{f_b\}_{b=1}^B$ where f_b outputs $f_b^n(t)$ as τ_n -level quantile estimate at time t ;
- 2: **Initialize:** $w_b^n(1) = \frac{1}{B}$, $\forall b \in [B], n \in [N]$;
- 3: **for** $t = 1, \dots, T$ **do**
- 4: **Output:** Output $q^H(t) := (q_n^H(t))_{n=1}^N$ where $q_n^H(t) = \sum_{b=1}^B w_b^n(t) f_b^n(t)$;
- 5: Update weights w_b^n for $b \in [B], n \in [N]$:

$$w_b^n(t+1) = \frac{w_b^n(t) \exp(-\eta \rho_n(f_b^n(t), Y_t))}{\sum_{b=1}^B w_b^n(t) \exp(-\eta \rho_n(f_b^n(t), Y_t))}$$

6: **end for**

This algorithms yields the following guarantee for each quantile level τ_n :

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \rho_n(q_n^H(t), Y_t) &\leq \frac{1}{T} \sum_{t=1}^T w_b^n(t) \rho_n(f_b^n(t), Y_t) \\ &\leq \min_b \frac{1}{T} \sum_{t=1}^T \rho_n(f_b^n(t), Y_t) + \frac{\eta}{T} \sum_{t=1}^T \sum_{b=1}^B w_b^n(t) \rho_n(f_b^n(t), Y_t)^2 + \frac{\log B}{\eta T}, \end{aligned}$$

where the first inequality is due to convexity of ρ_τ and the second inequality is due to the regret bound of the Hedge algorithm (Theorem C.4). This gives the omniprediction error bound for Algorithm 3 as follows:

$$\begin{aligned} &\sup_{\ell \in \mathcal{L}_{wql}, b \in [B]} \frac{1}{T} \sum_{t=1}^T [\ell(q^H(t), Y_t) - \ell(f_b(t), Y_t)] \\ &= \max_{n \in [N], b \in [B]} \frac{1}{T} \sum_{t=1}^T [\rho_n(q_n^H(t), Y_t) - \rho_n(f_b^n(t), Y_t)] \\ &\leq \eta \max_{n \in [N]} \left(\frac{1}{T} \sum_{t=1}^T \sum_{b=1}^B w_b^n(t) \rho_n(f_b^n(t), Y_t)^2 \right) + \frac{\log B}{\eta T} \\ &\leq \eta \max_{n \in [N]} \left(\frac{1}{T} \sum_{t=1}^T \max_b \rho_n(f_b^n(t), Y_t)^2 \right) + \frac{\log B}{\eta T} \end{aligned} \tag{C.14}$$

C.3.2 Two-player game based algorithm

In this subsection, we introduce the second algorithm that is based on two-player game that is similar to Algorithm 1 in Section 4.3 then provide the omniprediction error bound for this algorithm. In the process, we prove that the output from this algorithm does not exhibit quantile crossing. We denote the output from Algorithm 4 by $q^G(t) := (q_n^G(t))_{n=1}^N$ where the superscript G denotes it is from the two-player Game based algorithm.

We show that $(q_n^G(t))_{n=1}^N$ does not exhibit quantile crossing.

Proposition C.4. *Define a function $V_n(\theta)$ and a quantity V_n^* for $n \in [N-1]$ as*

$$\begin{aligned} V_n(\theta) &:= \sum_{s=1}^n \sum_{b=1}^B w_b^s \max(\bar{f}_b^s(t) - \theta, 0) + \sum_{s=n+1}^N \sum_{b=1}^B w_b^s \max(\theta - f_b^s(t), 0) \geq 0 \\ V_n^* &:= \min_{\theta \in \mathbb{R}} V_n(\theta). \end{aligned}$$

and $V_0^* = V_N^* = 0$. Let J_n^* be the set of θ s that minimizes the $V_n(\theta)$. Then, there exists a sorted sequence $f_{\min}(t) \leq \theta_0^* \leq \theta_1^* \leq \dots \leq \theta_{N-1}^* \leq \theta_N^* = f_{\max}(t)$ such that $\theta_n^* \in J_n^*$ for $n \in [N-1]$.

Algorithm 4 Two-player game based omniprediction for weighted quantile losses

- 1: **Data:** Data stream $(Y_t)_{t=1}^T$, quantile levels $\tau_1 < \tau_2 < \dots < \tau_N$, Learning rate $\eta > 0$, Base forecasters: $\{f_b\}_{b=1}^B$ where f_b outputs $f_b^n(t)$ as τ_n -level quantile estimate at time t ;
- 2: **Initialize:** $w_b^n(1) = \frac{1}{NB}$, $\forall n \in [N], b \in [B]$;
- 3: **for** $t = 1, \dots, T$ **do**
- 4: **1) Make prediction at time t :**
- 5: Define $V_0^* = V_N^* = 0$, $\theta_0^* = f_{\min}(t)$, $\theta_N^* = f_{\max}(t)$;
- 6: **for** $n = 1, \dots, N-1$ **do**
- 7: Define $V_n(\theta)$ as:

$$V_n(\theta) = \sum_{s=1}^n \sum_{b=1}^B w_b^s \max(f_b^s(t) - \theta, 0) + \sum_{s=n+1}^N \sum_{b=1}^B w_b^s \max(\theta - f_b^s(t), 0)$$

- 8: Compute V_n^* and θ_n^* :

$$V_n^* := \min_{\theta \in [f_{\min}(t), f_{\max}(t)]} V_n(\theta), \quad \theta_n^* = \min \{ \theta \mid \theta \in [\theta_{n-1}^*, f_{\max}(t)], V_n(\theta) = V_n^* \} \quad (\text{C.15})$$

- 9: **end for**
- 10:
- 11: **Output** $q^G(t) := (q_n^G(t))_{n=1}^N$ where

$$q_n^G(t) = \frac{1}{\sum_{b=1}^B w_b^n(t)} \left(\sum_{b=1}^B w_b^n(t) f_b^n(t) + V_{n-1}^* - V_n^* \right);$$

- 12: **2) Update the weights for each ρ_n and f_b :**

$$w_b^n(t+1) = \frac{w_b^n(t) \exp(\eta[\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t)])}{\sum_{n=1}^N \sum_{b=1}^B w_b^n(t) \exp(\eta[\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t)])}, \quad \forall n \in [N], b \in [B];$$

- 13: **end for**
-

Additionally, q_n defined as

$$q_n = \frac{1}{\sum_{b=1}^B w_b^n(t)} \left(\sum_{b=1}^B w_b^n(t) f_b^n(t) + V_{n-1}^* - V_n^* \right). \quad (\text{C.16})$$

satisfies $\theta_{n-1}^* \leq q_n \leq \theta_n^*$ for $n \in [N-1]$ and thus $f_{\min}(t) \leq q_1 \leq q_2 \dots \leq q_N \leq f_{\max}(t)$ holds.

Proof. Since $V_n(\theta)$ shares the same structure with $V_n(j)$ defined in equation (C.1), existence of a sorted sequence follows the same proof in Lemma C.1. To prove that such θ_n^* takes values in $[f_{\min}(t), f_{\max}(t)]$ for all $n \in [N-1]$, it suffices to show that $J_n^* \cap [f_{\min}(t), f_{\max}(t)] \neq \emptyset$ for all $n \in [N-1]$. On $\theta \leq f_{\min}(t)$, the second sum in $V_n(\theta)$ is simply zero and thus $V_n(\theta)$ is non-increasing. On $\theta \geq f_{\max}(t)$, the first sum in $V_n(\theta)$ is zero and thus $V_n(\theta)$ is non-decreasing. Therefore, J_n^* cannot only contain values that are not in $[f_{\min}(t), f_{\max}(t)]$.

Next, we prove that $\theta_{n-1}^* \leq q_n \leq \theta_n^*$ for $n \in [N-1]$. Observe the identity:

$$V_{n-1}(\theta) - V_n(\theta) = \sum_{b=1}^B w_b^n(t) (\max(\theta - f_b^n(t), 0) - \max(f_b^n(t) - \theta, 0)) = \sum_{b=1}^B w_b^n(t) (\theta - f_b^n(t)),$$

which implies that

$$q_n = \frac{1}{\sum_{b=1}^B w_b^n(t)} \left(\sum_{b=1}^B w_b^n(t) f_b^n(t) + V_{n-1}^* - V_{n-1}(\theta_n^*) + \sum_{b=1}^B w_b^n(t) (\theta_n^* - f_b^n(t)) \right) \leq \theta_n^*,$$

since $V_{n-1}^* - V_{n-1}(\theta_n^*) \leq 0$ by the definition of V_n^* . Similarly,

$$q_n = \frac{1}{\sum_{b=1}^B w_b^n(t)} \left(\sum_{b=1}^B w_b^n(t) f_b^n(t) + V_{n-1}(\theta_{n-1}^*) - V_n^* + \sum_{b=1}^B w_b^n(t) (\theta_{n-1}^* - f_b^n(t)) \right) \geq \theta_{n-1}^*,$$

which completes the proof. $\square$

Next, we show that $(q_n)_{n=1}^N$ are a solution to the minimax program .

Theorem C.3. *Given quantile levels $\tau_1 < \tau_2 < \dots < \tau_N$ and quantile forecasts $\hat{f}_b^n(t)$ for $b \in [B]$, $n \in [N]$, define V_n^* and θ_n^* for $n \in [N] \cap \{0\}$ as in Proposition C.4 and define p_n^* , $n \in [N]$ as in equation (C.16). Then, $P_n^* \equiv q_n$ for $n \in [N]$ with*

$$P_y^* = \tau_1 \delta_{f_{\min}(t)} + (1 - \tau_N) \delta_{f_{\max}(t)} + \sum_{n=1}^{N-1} (\tau_{n+1} - \tau_n) \delta_{\theta_n^*}$$

is a solution of the minimax program

$$\min_{P_1, P_2, \dots, P_N \in \mathcal{P}(\mathbb{R})} \max_{P_y \in \mathcal{P}(\mathbb{R})} \sum_{n=1}^N \sum_{b=1}^B w_b^n(t) \mathbb{E}_{q_n \sim P_n, Y \sim P_y} [\rho_n(q_n, Y) - \rho_n(f_b^n(t), Y)]$$

and the value of the minimax program is $-\sum_{n=1}^{N-1} (\tau_{n+1} - \tau_n) V_n^* \leq 0$.

Proof. Similar to the proof of Proposition C.2, define $O((P_n)_{n=1}^N, P_y)$ as

$$O((P_n)_{n=1}^N, P_y) = \sum_{n=1}^N \sum_{b=1}^B w_b^n(t) \mathbb{E}_{q_n \sim P_n, Y \sim P_y} [\rho_n(q_n, Y) - \rho_n(f_b^n(t), Y)].$$

It suffices to prove

$$O((P_n^*)_{n=1}^N, P_y^*) = \max_{P_y} O((P_n^*)_{n=1}^N, P_y) = \min_{P_n \in \mathcal{P}(\mathbb{R}), n \in [N]} O((P_n)_{n=1}^N, P_y^*).$$

Observe that $\rho_\tau(x, y) = \tau(y - x) + \max(x - y, 0)$. This simplifies $O((P_n)_{n=1}^N, P_y)$ to

$$\begin{aligned} & O((P_n)_{n=1}^N, P_y) \\ &= \mathbb{E}_{Y \sim P_y} \sum_{n=1}^N \sum_{b=1}^B w_b^n(t) [\tau_n(Y - q_n) + \max(q_n - Y, 0) - \tau_n(Y - f_b^n(t)) - \max(f_b^n(t) - q_n, 0)] \\ &= \mathbb{E}_{Y \sim P_y} \sum_{n=1}^N \sum_{b=1}^B w_b^n(t) [\max(q_n - Y, 0) - \max(f_b^n(t) - q_n, 0)] + C, \end{aligned}$$

where $C = -\sum_{n=1}^N (V_n^* - V_{n-1}^*) \tau_n = -\sum_{n=1}^{N-1} (\tau_{n+1} - \tau_n) V_n^*$ does not depend on P_y . Now we divide cases into $y < q_1$, $q_1 \leq y < q_2$, ..., $q_{N-1} \leq y < q_N$.

$$\begin{aligned} & \sum_{n=1}^N \sum_{b=1}^B w_b^n(t) [\max(q_n - y, 0) - \max(f_b^n(t) - q_n, 0)] \\ &= \begin{cases} -\sum_{n=1}^N \sum_{b=1}^B w_b^n \max(y - f_b^n(t), 0) & y < q_1 \\ -(\sum_{s=n+1}^N \sum_{b=1}^B w_b^s \max(y - f_b^s(t), 0) \\ \quad + \sum_{s=1}^n \sum_{b=1}^B w_b^s \max(f_b^s(t) - y, 0)) + V_n^* & y \in [q_n, q_{n+1}), n \in [N-1] \\ -\sum_{n=1}^N \sum_{b=1}^B w_b^n \max(f_b^n(t) - y, 0) & y \geq q_N \end{cases} \end{aligned}$$

In the interval $(-\infty, q_1)$, the quantity is maximized at $y = \theta_0^* = f_{\min}(t)$ and its maximum value is 0. In interval $[q_n, q_{n+1})$, the quantity is maximized at $y = \theta_n^*$ (contained in the interval as shown in Proposition C.4) and its maximum value is 0. In interval (q_{N-1}, ∞) , the quantity is maximized at $y = \theta_N^* = f_{\max}(t)$ and its maximum value is 0. Since P_y^* only puts mass on the points $\{\theta_n^*\}_{n=0}^N$, $O((P_n^*)_{n=1}^N, P_y^*) = \max_{P_y} O((P_n^*)_{n=1}^N, P_y) = C = -\sum_{n=1}^{N-1} (\tau_{n+1} - \tau_n) V_n^* \leq 0$. Next, $O((P_n^*)_{n=1}^N, P_y^*) = \min_{P_n \in \mathcal{P}(\mathbb{R}), n \in [N]} O((P_n)_{n=1}^N, P_y^*)$ follows directly from the propriety of ρ_τ and the fact that q_n is τ_n -level quantile of P_y^* . $\square$

Putting all these together, the omniprediction error bound for Algorithm 4 is as following:

$$\begin{aligned} & \sup_{\ell \in \mathcal{L}_{wql}, b \in [B]} \frac{1}{T} \sum_{t=1}^T [\ell(q^G(t), Y_t) - \ell(f_b(t), Y_t)] \\ & \leq \eta \left(\frac{1}{T} \sum_{t=1}^T \max_{n,b} (\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t))^2 \right) + \frac{\log BN}{\eta T}. \end{aligned} \quad (\text{C.17})$$

Proof.

$$\begin{aligned}
& \sup_{\ell \in \mathcal{L}_{wql}, b \in [B]} \frac{1}{T} \sum_{t=1}^T [\ell(q^G(t), Y_t) - \ell(f_b(t), Y_t)] \\
&= \max_{n \in [N], b \in [B]} \frac{1}{T} \sum_{t=1}^T [\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t)] \\
&\leq \frac{1}{T} \sum_{t=1}^T \sum_{n=1}^N \sum_{b=1}^B w_b^n(t) [\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t)] \\
&\quad + \frac{\eta}{T} \sum_{t=1}^T \sum_{n=1}^N \sum_{b=1}^B w_b^n(t) (\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t))^2 + \frac{\log BN}{\eta T} \\
&\stackrel{\text{Thm. C.3}}{\leq} \eta \left(\frac{1}{T} \sum_{t=1}^T \max_{n,b} (\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t))^2 \right) + \frac{\log BN}{\eta T}.
\end{aligned}$$

□

C.3.3 Comparison between two algorithms

We conclude the section by comparing the two algorithms in terms of time complexity, quantile crossing, and omniprediction error bound.

Time complexity The time complexity of Algorithm 3 is $O(NB)$ per time step. Algorithm 4 is more complicated, where the main computational bottleneck lies in computing V_n^* and θ_n^* for each $n \in [N - 1]$ (see Equation (C.15)). A naive computation of the minimizers $\{\theta_n^*\}$ requires $O(N^3 B^2)$ per each time step. Appendix C.4.2 introduces a subroutine that computes all $\{\theta_n^*\}$ and $\{V_n^*\}$ in $O(K \log K + NB)$ time, where K is the number of unique values in $\{f_b^n(t)\}_{b,n}$, by exploiting the structure of $\{V_n^*\}$.

Quantile crossing. Due to Proposition C.4, quantile forecasts from Algorithm 4 does not exhibit quantile crossing. On the other hand, Algorithm 3 does not guarantee such property and post-sorting the resulting quantile does not always decrease the loss. Appendix C.6.2 provides examples where post-sorting rather increases the loss.

Omniprediction error bound. Comparing the omniprediction error bound (C.14) and (C.17), we notice that the former bound is a function of the error of base forecasters, $\rho_n(f_b^n(t), Y_t)$, while the latter bound is a function of relative error of two-player game based algorithm output with respect to base forecasters, $\rho_n(q_n^G(t), Y_t) - \rho_n(f_b^n(t), Y_t)$. This suggests that the two-player based algorithm would achieve a better omniprediction error for harder prediction tasks since the error of base forecasters is likely to be larger compared to the relative error.

C.4 Additional algorithmic details

Algorithm 5 describes the full description of our main omniprediction algorithm, Algorithm 1. Appendix C.4.1 discusses an efficient subroutine for computing V_n^* and j_n^* , defined in equations (C.18). Appendix C.4.2 discusses an efficient subroutine for computing θ_n^* , defined in equations (C.15).

C.4.1 Efficient subroutine for Algorithm 5

This subsection introduces Algorithm 6, an efficient subroutine to compute V_n^* and j_n^* for each $n \in [N - 1]$ in $O(mNB)$ time. Recall the definition of V_n^* from Lemma C.1:

$$V_n^* := \min_{j \in [m]_0} \left(\sum_{i=j+1}^m \sum_{s=1}^n \sum_{b=1}^B w_{ib}^s \mathbb{1}\{\theta_i < f_b^s(t)\} + \sum_{i=1}^j \sum_{s=n+1}^N \sum_{b=1}^B w_{ib}^s (1 - \mathbb{1}\{\theta_i < f_b^s(t)\}) \right)$$

C.4.2 Efficient subroutine for Algorithm 4

This subsection introduces Algorithm 7 that computes $\{\theta_n^*\}_{n=0}^N$ in $O(K \log(K))$ time, where K is number of unique values in $\{f_b^n(t)\}_{b,n}$. With this algorithm, now the computational bottleneck lies in computing $V_n^* = V_n(\theta_n^*)$ for each $n \in [N - 1]$, which takes $O(N^2B)$ in total. To reduce the time complexity even further, we introduce Algorithm 8 that computes $V_{n-1}^* - V_n^*$ for all $n \in [N - 1]$, which is all we need to compute the minimax solution $q_n^G(t)$ (See Proposition C.4), in $O(NB)$ time.

We begin with introducing an efficient algorithm to compute $\{\theta_n^*\}_{n=0}^N$ in $O(K \log(K))$ time. Recall the definition of V_n^* in Proposition C.4:

$$V_n(\theta) := \sum_{s=1}^n \sum_{b=1}^B w_b^s \max(f_b^s(t) - \theta, 0) + \sum_{s=n+1}^N \sum_{b=1}^B w_b^s \max(\theta - f_b^s(t), 0), \quad V_n^* := \min_{\theta \in \mathbb{R}} V_n(\theta).$$

The key point in efficiently computing V_n^* is that $V_n(\theta)$ is a sum of piecewise linear functions (hinge functions) in θ . This implies that the minimizer θ_n^* is the point where the slope of $V_n(\theta)$ changes from negative to positive. Another key observation is that the slope of $V_n(\theta)$ only changes at $\{f_b^s(t)\}_{b,s}$ and it increases by w_b^s . More precisely, the change of slope at $f_b^s(t)$ is $\sum_{\{(x,z): f_x^z(t)=f_b^s(t)\}} w_x^z$. It is important to note that the amount of change at each $f_b^s(t)$ is same for all $V_n(\theta)$, $n \in [N - 1]$.

Then, we observe that the slope of $V_n(\theta)$ at $\theta < \min_{b,n} f_b^n(t)$ is $-\sum_{s=1}^n \sum_{b=1}^B w_b^s$. This implies that finding θ_n^* reduces to find the smallest $f_b^s(t)$ such that cumulative slope increase is larger than or equal to $\sum_{s=1}^n \sum_{b=1}^B w_b^s$. Formally, θ_n^* equals to the smallest $f_b^s(t)$ such that $\sum_{\{(x,z): f_x^z(t) \leq f_b^s(t)\}} w_x^z \geq \sum_{s=1}^n \sum_{b=1}^B w_b^s$.

Algorithm 5 Full description of Algorithm 1

```

1: Data: Data stream  $(Y_t)_{t=1}^T$ , quantile levels  $\tau_1 < \tau_2 < \dots < \tau_N$ , Grid size  $m \in \mathbb{N}$ , Learning
   rate  $\eta > 0$ , Base forecasters:  $\{f_b\}_{b=1}^B$  where  $f_b$  outputs  $f_b^n(t)$  as  $\tau_n$ -level quantile estimate
   at time  $t$ ;
2: Initialize:  $w_{ib}^n(1) = \frac{1}{mNB}$ ,  $\forall i \in [m], n \in [N], b \in [B]$ ;
3: for  $t = 1, \dots, T$  do
4:   1) Make prediction at time  $t$ :
5:   Define  $V_0^* = V_N^* = 0$ ,  $j_0^* = 0$ ,  $j_N^* = m$ ;
6:   Define  $w_i^n = (w_{ib}^n(t))_{b=1}^B$ ,  $\mathbf{1}_i^n = (\mathbf{1}\{\theta_i < f_b^n(t)\})_{b=1}^B$  and  $\mathbf{1}_B = (1, \dots, 1) \in \mathbb{R}^B$  for  $\forall i, n$ .
7:   for  $n = 1, \dots, N - 1$  do
8:     Define  $V_n(j)$  as  $V_n(j) = \sum_{i=j+1}^m \sum_{s=1}^n w_i^s \cdot \mathbf{1}_i^s + \sum_{i=1}^j \sum_{s=n+1}^N w_i^s \cdot (\mathbf{1}_B - \mathbf{1}_i^s)$ 
9:     Compute  $V_n^*$  and  $j_n^*$ :

$$V_n^* = \min_{j \in [m]} V_n(j), \quad j_n^* = \min \{j \mid j \in [j_{n-1}^*, m], V_n(j) = V_n^*\} \quad (\text{C.18})$$

10:   end for
11:
12:   for  $n = 1, \dots, N$  do
13:     if  $V_n(j_{n-1}^*) = V_n(j_n^*)$  then
14:        $\hat{P}_n(t) = \delta_{j_{n-1}^*/m}$ ;
15:     else
16:       Define  $B_k := \sum_{i=1}^k w_i^n \cdot \mathbf{1}_B - \sum_{i=1}^m w_i^n \cdot \mathbf{1}_i^n$ 
17:       Find  $j_{n-1}^* \leq k_n < j_n^*$  s.t.  $B_{k_n} < V_{n-1}^* - V_n^* \leq B_{k_n+1}$  (Existence proved in Prop.
C.1)
18:        $\hat{P}_n(t) = (1 - C_n)\delta_{k_n/m} + C_n\delta_{((k_n+1)/m)}$ , where  $C_n = (V_{n-1}^* - V_n^* - B_{k_n}) / (w_{k_n}^n \cdot \mathbf{1}_B)$ 
19:     end if
20:   end for
21:   Output  $\hat{P}(t) = (\hat{P}_n(t))_{n=1}^N$ ;
22:
23:   2) Update the weights for each  $S_{n,i}$  and  $f_b$ :

$$w_{ib}^n(t+1) = \frac{w_{ib}^n(t) \exp(\eta(\mathbb{E}_{q_n \sim \hat{P}_n(t)}[\ell_{\tau_n, \theta_i}(q_n, Y_t)] - \ell_{\tau_n, \theta_i}(\hat{f}_b^n(t), Y_t)))}{\sum_{i,n,b} w_{ib}^n(t) \exp(\eta(\mathbb{E}_{q_n \sim \hat{P}_n(t)}[\ell_{\tau_n, \theta_i}(q_n, Y_t)] - \ell_{\tau_n, \theta_i}(\hat{f}_b^n(t), Y_t)))}, \forall i, n, b$$

24: end for

```

Algorithm 6 Efficient computation of V_n and its minimizer j_n^*

-
- 1: **Input:** Weights $(w_{ib}^s)_{s \in [N], i \in [m], b \in [B]}$, indicators $(\mathbb{1}\{\theta_i < f_b^s(t)\})_{s \in [N], i \in [m], b \in [B]}$
 - 2: **Output:** Values $\{V_n\}_{n=1}^{N-1}$ and minimizers $\{j_n^*\}_{n=1}^{N-1}$
 - 3: 1) **Compute aggregated weights and indicators:**

$$w_i^s = \sum_{b=1}^B w_{ib}^s, \quad I_i^s = \sum_{b=1}^B w_{ib}^s \mathbb{1}\{\theta_i < f_b^s(t)\}, \quad \forall i \in [m], s \in [N].$$

- 4: 2) **Compute cumulative sums:**

$$\text{FrontSum}[n, i] = \sum_{s=1}^n I_i^s, \quad \text{BackSum}[n, i] = \sum_{s=n+1}^N (w_i^s - I_i^s), \quad \forall i \in [m], n \in [N-1].$$

- 5: 3) **Find j_n^* for each $n \in [N-1]$:**
- 6: Define $j_0^* = 0$. Initialize $\text{cost}[j] = 0, \forall j \in [m]_0$.
- 7: **for** $n = 1, \dots, N-1$ **do**
- 8: Compute the cost at $j = j_{n-1}^*$:

$$\text{cost}[j_{n-1}^*] = \sum_{i=j_{n-1}^*+1}^m \text{FrontSum}[n, i] + \sum_{i=1}^{j_{n-1}^*} \text{BackSum}[n, i], \quad V_n \leftarrow \text{cost}[j_{n-1}^*]$$

- 9: **for** $j = j_{n-1}^* + 1, \dots, m$ **do**
 - 10: $\text{cost}[j] = \text{cost}[j-1] - \text{FrontSum}[n, j-1] + \text{BackSum}[n, j-1]$
 - 11: $V_n \leftarrow \min(V_n, \text{cost}[j])$
 - 12: **end for**
 - 13: $j_n^* = \arg \min_{j_{n-1}^* \leq j \leq m} \text{cost}[j]$
 - 14: **end for**
 - 15: **Return:** $\{V_n\}_{n=1}^{N-1}$ and $\{j_n^*\}_{n=1}^{N-1}$
-

Next, we introduce an algorithm that computes $V_{n-1}^* - V_n^*$ for all $n \in [N-1]$ in $O(NB)$ time. Note that naive computation of V_n^* using θ_n^* takes $O(NB)$ time and thus $O(N^2B)$ in total.

Algorithm 7 Efficient computation of $\{\theta_n^*\}_{n=0}^N$

-
- 1: **Input:** Weights $(w_b^n)_{n \in [N], b \in [B]}$, forecasts $\{f_b^n(t)\}_{n \in [N], b \in [B]}$
 - 2: **1) Compute cumulative row sums:**
 - 3: Define $W[n] = \sum_{s=1}^n \sum_{b=1}^B w_b^s$, $n \in [N]$.
 - 4: **2) Compute the slope increases at each $f_b^s(t)$:**
 - 5: Flatten $\{(w_b^n, f_b^n(t))\}_{b,n}$ into pairs $\{(\tilde{w}_k, \tilde{f}_k)\}_{k=1}^{NB}$ and sort them by $\tilde{f}_k$ in nondecreasing order.
 - 6: Group equal forecast values and let the distinct sorted values be $\{g_k\}_{k=1}^K$ with $g_1 < g_2 < \dots < g_K$.
 - 7: For each $k \in [K]$, compute the sum of corresponding weights: $v_k = \sum_{\{(b,n): f_b^n(t)=g_k\}} w_b^n$.
 - 8: Compute the cumulative sum of v_k : $Vcum[k] = \sum_{\ell=1}^k v_\ell$.
 - 9: **3) Find θ_n^* for each $n \in [N-1]$:**
 - 10: **for** $n = 1, \dots, N-1$ **do**
 - 11: Binary search on $Vcum$ (sorted b/c $v_k > 0$) to find smallest k_n^* such that $Vcum[k_n^*] \geq W[n]$.
 - 12: $\theta_n^* \leftarrow g_{k_n^*}$.
 - 13: **end for**
-

Before introducing the algorithm, we observe what $V_{n-1}^* - V_n^*$ is.

$$\begin{aligned}
V_{n-1}(\theta_{n-1}^*) - V_n(\theta_n^*) &= \sum_{s=1}^{n-1} \sum_{b=1}^B w_b^s (\max(f_b^s(t) - \theta_{n-1}^*, 0) - \max(f_b^s(t) - \theta_n^*, 0)) \\
&\quad + \sum_{s=n+1}^N \sum_{b=1}^B w_b^s (\max(\theta_{n-1}^* - f_b^s(t), 0) - \max(\theta_n^* - f_b^s(t), 0)) \\
&\quad + \sum_{b=1}^B w_b^n (\max(\theta_{n-1}^* - f_b^n(t), 0) - \max(f_b^n(t) - \theta_n^*, 0))
\end{aligned}$$

Each component results in different values depending on whether $f_b^n(t) < \theta_{n-1}^*$ or $\theta_{n-1}^* \leq f_b^n(t) < \theta_n^*$ or $\theta_n^* \leq f_b^n(t)$. Fortunately, some calculations on the quantity above (divide cases etc.) gives the following simple formula:

$$\begin{aligned}
&V_{n-1}(\theta_{n-1}^*) - V_n(\theta_n^*) \\
&= \sum_{s=1}^N \sum_{b=1}^B w_b^s (\theta_{n-1}^* \mathbb{1}\{f_b^s(t) < \theta_{n-1}^*\} + f_b^s(t) \mathbb{1}\{\theta_{n-1}^* \leq f_b^s(t) < \theta_n^*\} + \theta_n^* \mathbb{1}\{\theta_n^* \leq f_b^s(t)\}) \\
&\quad - \theta_{n-1}^* \left(\sum_{s=1}^{n-1} \sum_{b=1}^B w_b^s \right) - \theta_n^* \left(\sum_{s=n+1}^N \sum_{b=1}^B w_b^s \right) - \sum_{b=1}^B w_b^n f_b^n(t).
\end{aligned}$$

Observe that most quantities can be precomputed in $O(NB)$ time. Algorithm 8 shows the details.

Algorithm 8 Efficient computation of $V_{n-1}^* - V_n^*$ for all $n \in [N - 1]$

- 1: **Input:** Weights $(w_b^n)_{b,n}$, forecasts $\{f_b^n(t)\}_{b,n}$, $\{g_k\}_{k=1}^K$, $\{\theta_n^* = g_{k_n^*}\}_{n=0}^N$ from Algorithm 7
- 2: **1) Compute cumulative row sums:**
- 3: Define $W[n] = \sum_{s=1}^n \sum_{b=1}^B w_b^s$, $n \in [N]$, $W[0] = 0$.
- 4: Define $WB[n] = \sum_{s=n+1}^N \sum_{b=1}^B w_b^s$, $n \in [N - 1]$, $WB[N] = 0$.
- 5: Define $V[k] = \sum_{\{(b,n): f_b^n(t)=g_k\}} w_b^n$, $k \in [K]$.
- 6: Define $VF[k] = \sum_{\{(b,n): f_b^n(t)=g_k\}} w_b^n * f_b^n(t)$, $k \in [K]$.
- 7: Define $Vcum[k] = \sum_{\ell=1}^k V[\ell]$, $k \in [K]$, $Vcum[0] = 0$.
- 8: Define $VFcum[k] = \sum_{\ell=1}^k VF[\ell]$, $k \in [K]$, $VFcum[0] = 0$.
- 9: **2) Compute $V_{n-1}^* - V_n^*$ for each $n \in [N - 1]$:**
- 10: **for** $n = 1, \dots, N - 1$ **do**

$$\begin{aligned}
V_{n-1}^* - V_n^* &= \theta_{n-1}^*(Vcum[\max(k_{n-1}^*, 0)]) + (VFcum[k_n^* - 1] - VFcum[\max(k_{n-1}^*, 0)]) \\
&\quad + \theta_n^*(1 - Vcum[k_n^* - 1]) - \theta_{n-1}^*(W[n - 1]) - \theta_n^*(WB[n]) - \sum_{b=1}^B w_b^n f_b^n(t).
\end{aligned}$$

- 11: **end for**
 - 12: **Output:** $\{V_{n-1}^* - V_n^*\}_{n=1}^{N-1}$
-

C.5 Additional empirical results

C.5.1 Experiment details

During the pandemic, the United States Center for Disease Control and Prevention solicited forecasts of future disease outcomes from research groups throughout the country [CHW⁺22]. In our experiment, we focus in particular on forecasts of the future number of people that will be hospitalized with Covid-19 on a given day. Below is the detailed information about the data.

- Horizon: 1, 2, 3, 4 weeks ahead
- Location (total 52): 51 state-level series (50 states + DC) and the U.S. aggregate series
- Dates: 2020-12-29 to 2023-06-05 (127 weeks)
- Base forecasters (total 18): CU-select, GT-DeepCOVID, COVIDhub-baseline, Karlenpypm, JHU_IDD-CovidSP, MOBS-GLEAM_COVID, USC-SI_kJalpha, JHUAPL-Bucky, JHUAPL-Gecko, MUNI-ARIMA, CMU-TimeSeries, UMass-trends_ensemble, PSI-DICE, IHME-CurveFit, CUB_PopCouncil-SLSTM, BPagano-RtDriven, LANL-GrowthRate, UVA-Ensemble

- Ensemble models (total 2): COVIDhub-4_week_ensemble (COVIDhub-Ens), JHUAPL-SLPHospEns (JHUAPL-SLPEns)
- Quantile levels (total 23): [0.01, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.975, 0.99]

We choose the grid spacing for each region so that the resulting discretization yields sufficiently fine resolution across time series based on their respective ranges. In particular, this choice ensures more than 500 grid points for all $208 = 52 \times 4$ time series, and more than 1,000 grid points for over 150 time series. The corresponding grid spacing used for each region is as follows:

- RI, ND, AK, VT, WY: 2
- NM, ID, NE, WV, HI, NH, ME, DE, SD: 5
- OR, UT, NV, IA, KS, MS, MT, DC: 10
- VA, WA, AZ, TN, MA, IN, MO, MD, CO, WI, MN, SC, AL, LA, KY, OK, CT, AR: 20
- PA, IL, OH, GA, NC, MI, NJ: 50
- CA, TX, FL, NY: 100
- US: 500

C.5.2 Additional plots

Figure C.1 compares quantile forecasts for US-aggregate hospitalizations. During the pandemic, COVIDhub-Ens was constructed as a weighted average of the quantile estimates of the base forecasters, whereas JHUAPL-SLPEns was obtained by first constructing predictive distributions and then averaging the resulting CDFs [Mul24]. Hedge-QL also aggregates quantile estimates via weighted averaging. It is well-known that averaging quantile forecasts produces sharper intervals than averaging predictive distributions [LJGCW13]. Consistent with this, COVIDhub-Ens and Hedge-QL produce relatively sharp prediction bands, whereas JHUAPL-SLPEns yields wider intervals. Interestingly, although our algorithm only uses quantile forecasts, it produces intervals that are slightly wider than those of COVIDhub-Ens and Hedge-QL, yet noticeably sharper than JHUAPL-SLPEns, while still closely tracking the observed trajectory.

Analogous to Figure 4.4, Figure C.2 shows how the performance of Hedge-QL varies with the learning rate. The figure shows the realized total omni-error and total quantile loss realized by Hedge-QL for 1- to 4-week-ahead horizons on the US-aggregate data with learning rate constants $c_H \in \{10^{i/4} \mid i = -4, \dots, 12\}$.

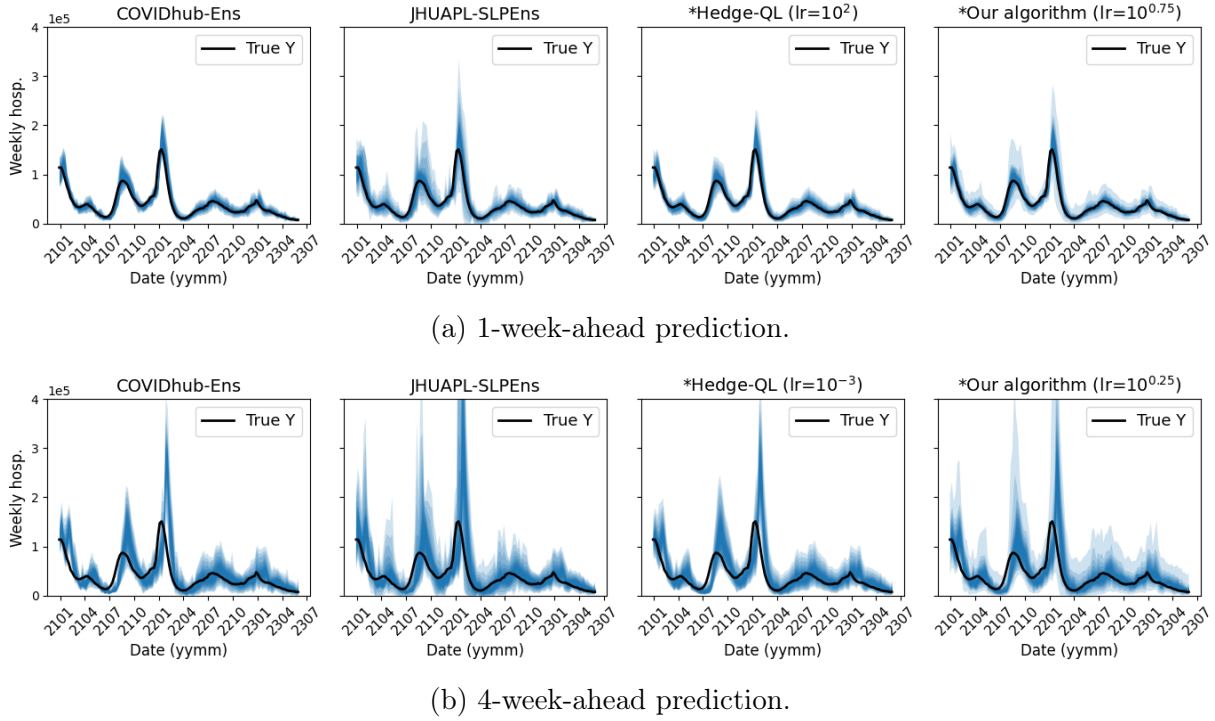


Figure C.1: Quantile forecasts for US-aggregate weekly COVID-19 hospitalizations at 1-week-ahead and 4-week-ahead horizons. The learning rate for ***Hedge-QL** and ***Our algorithm** is selected to achieve the best total omni-error. The black curve shows the observed hospitalizations. The blue shaded regions show nested quantile prediction bands: darker regions correspond to more central quantile intervals, while lighter regions correspond to wider intervals.

Lastly, Figure C.3 summarizes performance of all algorithms including 18 base forecasters using standardized ranks. Models in the column of the figure are ordered by their first-quartile performance (lowest quartile of standardized rank), with higher placement indicating better performance. Besides ***Our algorithm** and ***Hedge-QL**, multiple instances of our algorithm with a fixed learning rate are ranked higher than COVIDHUB-Ens and JHUAPL-SLPEns as well as all of the base forecasters. In particular, the both the first and third quartile of standardized ranks of all instances of our algorithm are comparable to or higher than the two ensemble models. This implies that our method, for a wide range of learning rates, can be viewed as better and more robust model compared to existing ensemble models, COVIDHUB-Ens and JHUAPL-SLPEns.

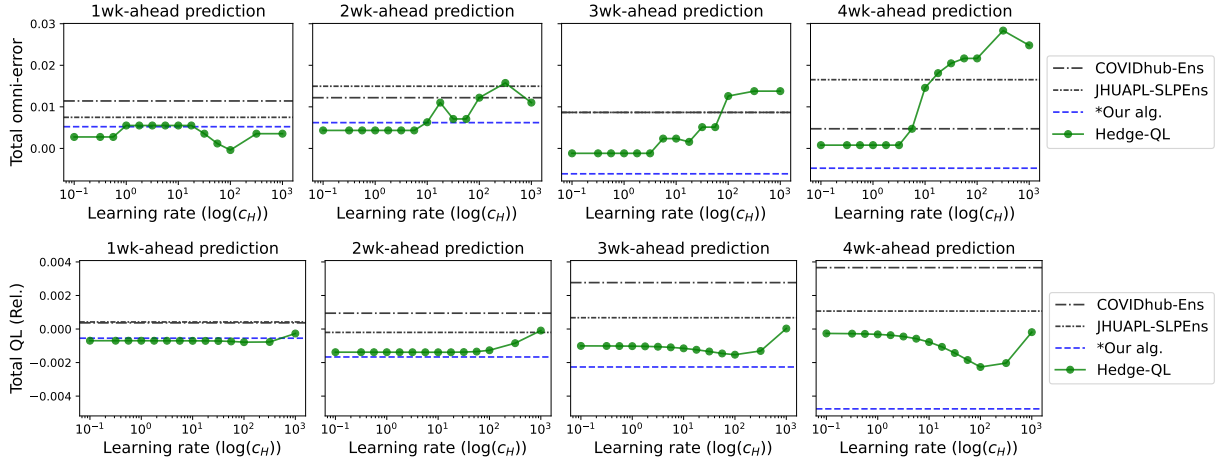


Figure C.2: Sensitivity of Hedge-QL to the choice of learning rate. Each panel shows the performance of Hedge-QL as a function of the learning rate (green curve) as measured by the total omni-error (top row) or total quantile loss relative to best base forecaster (bottom row) on the US-agg data. Panels from left to right shows results at one to four week time horizons, while horizontal lines indicate the performance of **Our algorithm* and the two baseline ensembles. Lower values indicate better performance.

C.6 Miscellaneous additional details

C.6.1 Regularity conditions

This subsection provides details of the regularity conditions in Theorem 4.2. In Proposition 4.2 in [FZ16], the conditions are stated in terms of the identification function V and its expectation over Y , $\bar{V}$ (see Definition 2.2 in [FZ16]). For the τ -level quantile functional, the identification function on $q, y \in [0, 1]$ and F a cdf supported on $[0, 1]$ is as follows:

$$V(q, y) = \mathbb{1}(y \leq q) - \tau \quad \text{and} \quad \bar{V}(q, F) := \mathbb{E}_{Y \sim F} V(q, Y) = F(q) - \tau.$$

For multiple quantile levels, $\{\tau_n\}_{n=1}^N$, the identification function is as follows:

$$\begin{aligned} V((q_n)_{n=1}^N, y) &= (\mathbb{1}(y \leq q_1) - \tau_1, \mathbb{1}(y \leq q_2) - \tau_2, \dots, \mathbb{1}(y \leq q_N) - \tau_N), \\ \bar{V}((q_n)_{n=1}^N, F) &= (F(q_1) - \tau_1, F(q_2) - \tau_2, \dots, F(q_N) - \tau_N). \end{aligned}$$

We now state the regularity conditions for Theorem 4.2. The conditions are referred to as (V1), (V3), (V4), (VS1), (S2), and (F1) in [FZ16]. Among these, it is straightforward to check that (V1), (V3), (V4), and (F1) hold for quantile functionals. We state the remaining conditions (S2) and (VS1), which are conditions on the loss function, below.

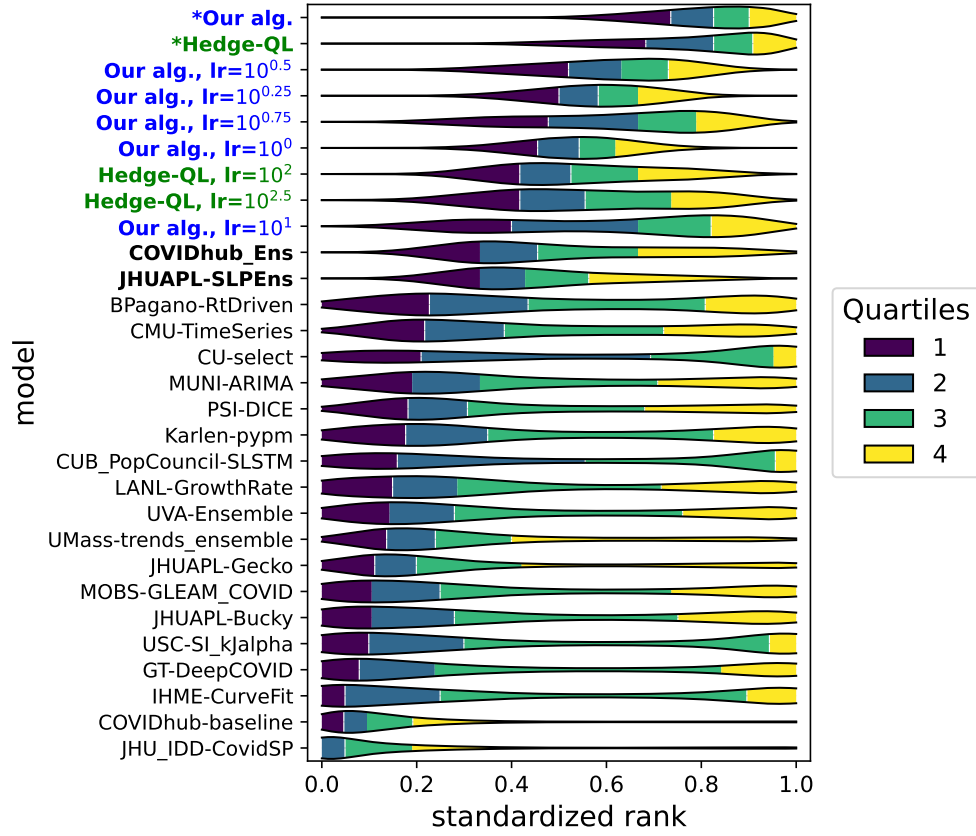


Figure C.3: Distribution of standardized ranks of instantaneous quantile loss across all (horizon, location, time) tuples for the 51 state-level series. For each tuple, we rank the available forecasts by quantile loss and normalize the ranks to $[0, 1]$. The colored segments represent the empirical distribution of ranks grouped into quartiles.

Assumption (VS1) The complement of the set

$$C := \{((x_n)_{n=1}^N, y) \in \mathbb{R}^N \times \mathbb{R} \mid \ell((x_n)_{n=1}^N, \cdot) \text{ are continuous at } y\}$$

has $(N + 1)$ -dimensional Lebesgue measure zero.

Assumption (S2) For every continuously differentiable cumulative distribution function F supported on $[0, 1]$, the function $\bar{\ell}(\cdot, F) := \mathbb{E}_{Y \sim F} \ell(\cdot, Y)$ is continuously differentiable, and its gradient is locally Lipschitz continuous. Furthermore, $\bar{\ell}(\cdot, F)$ is twice continuously differentiable at $(x_n)_{n=1}^N = (q_{\tau_n}(F))_{n=1}^N$.

C.6.2 The effect of post-sorting on proper loss functions

[FKM⁺23] showed that post-sorting never increases the loss for **unweighted** sums of pinball losses at different quantile levels. Unfortunately, the same does not hold for **weighted** sums of pinball losses. We present two results in this section. First, for a class of weighted sums of pinball losses at two quantile levels, we fully characterize the conditions under which post-sorting always decreases the loss. Conversely, when these conditions fail, there exists an instance (Y, x_1, x_2) for which post-sorting increases the loss. Second, we present a subclass of loss functions where post-sorting does not increase the loss.

Define the pinball loss at quantile level τ as $\rho_\tau(x) := \tau x \mathbf{1}\{x \geq 0\} + (1 - \tau)|x| \mathbf{1}\{x < 0\} = (\tau - \mathbf{1}\{x < 0\})x$.

Proposition C.5. *For $(\alpha, \beta) \in (0, 1)^2$ with $\alpha < \beta$, define a weighted quantile loss as*

$$\ell(x_1, x_2) := \mathbb{E}_Y[w_\alpha \rho_\alpha(Y - x_1) + w_\beta \rho_\beta(Y - x_2)].$$

If (w_α, w_β) satisfies one of the following conditions, then post-sorting does not increase the loss:

1. $w_\alpha = w_\beta$
2. $w_\alpha < w_\beta$ and $(1 - \alpha)w_\alpha \geq (1 - \beta)w_\beta$
3. $w_\alpha > w_\beta$ and $\beta w_\beta \geq \alpha w_\alpha$.

Conversely, if (w_α, w_β) does not satisfy any of the above conditions, then there exists (Y, x_1, x_2) such that post-sorting increases the loss.

Proof. Choose any $x_1 > x_2$. The loss before post-sorting is $\ell(x_1, x_2)$ and the loss after post-sorting is $\ell(x_2, x_1)$.

$$\begin{aligned} & \ell(x_1, x_2) - \ell(x_2, x_1) \\ &= w_\alpha \mathbb{E}_Y [(\alpha - \mathbf{1}\{Y < x_1\})(Y - x_1) - (\alpha - \mathbf{1}\{Y < x_2\})(Y - x_2)] + \\ & \quad w_\beta \mathbb{E}_Y [(\beta - \mathbf{1}\{Y < x_2\})(Y - x_2) - (\beta - \mathbf{1}\{Y < x_1\})(Y - x_1)] \\ &= w_\alpha E_Y [-\alpha(x_1 - x_2) + \mathbf{1}\{Y < x_1\}x_1 - \mathbf{1}\{Y < x_2\}x_2 - \mathbf{1}\{x_2 \leq Y < x_1\}Y] + \\ & \quad w_\beta E_Y [\{\beta(x_1 - x_2) + \mathbf{1}\{Y < x_2\}x_2 - \mathbf{1}\{Y < x_1\}x_1 + \mathbf{1}\{x_2 \leq Y < x_1\}Y\}] \\ &= \mathbb{E}_Y [-\alpha(x_1 - x_2) + \mathbf{1}\{x_2 \leq Y < x_1\}(x_1 - Y) + \mathbf{1}\{Y < x_2\}(x_1 - x_2)] w_\alpha + \\ & \quad \mathbb{E}_Y [\{\beta(x_1 - x_2) - \mathbf{1}\{x_2 \leq Y < x_1\}(x_1 - Y) - \mathbf{1}\{Y < x_2\}(x_1 - x_2)\}] w_\beta \end{aligned}$$

Denote the coefficient of w_α, w_β as c_α, c_β respectively. Notice that $c_\alpha + c_\beta = (\beta - \alpha)(x_1 - x_2) > 0$. Therefore $\ell(x_1, x_2) - \ell(x_2, x_1) > 0$ if $w_\alpha = w_\beta$ (Condition I).

Now, we consider Condition II: $w_\alpha < w_\beta$ and $(1 - \alpha)w_\alpha \geq (1 - \beta)w_\beta$. Denote $A = \mathbb{P}(Y < x_2)$, $B = \mathbb{P}(x_2 \leq Y < x_1)$ and observe that $\mathbb{E}_Y \mathbb{1}\{x_2 \leq Y < x_1\}(x_1 - Y) \in [0, B(x_1 - x_2)]$. Then the difference in loss is written as:

$$\begin{aligned} \ell(x_1, x_2) - \ell(x_2, x_1) &= (w_\alpha(A - \alpha) + w_\beta(\beta - A))(x_1 - x_2) - (w_\beta - w_\alpha)\mathbb{E}_Y \mathbb{1}\{x_2 \leq Y < x_1\}(x_1 - Y) \\ &\geq -(w_\beta - w_\alpha)A + \beta w_\beta - \alpha w_\alpha - (w_\beta - w_\alpha)B(x_1 - x_2) \\ &\geq -(w_\beta - w_\alpha)A + (w_\beta - w_\alpha) - (w_\beta - w_\alpha)B(x_1 - x_2) \\ &\geq (w_\beta - w_\alpha)(1 - A - B)(x_1 - x_2) \geq 0. \end{aligned}$$

Note that the second inequality used $\beta w_\beta - \alpha w_\alpha \geq w_\beta - w_\alpha$ which follows from Condition II and $A + B = \mathbb{P}(Y < x_1) \leq 1$. Condition III can be proved in a similar manner.

Next, we prove that there exists (Y, x_1, x_2) such that post-sorting increases the loss if (w_α, w_β) does not satisfy any of three conditions. First consider the case $w_\alpha < w_\beta$ and $(1 - \alpha)w_\alpha < (1 - \beta)w_\beta$. Let $Y \equiv x_2$ almost surely. Then $A = 0$, $B = 1$, and $\mathbb{E}_Y \mathbb{1}\{x_2 \leq Y < x_1\}(x_1 - Y) = x_1 - x_2$. Therefore,

$$\begin{aligned} \ell(x_1, x_2) - \ell(x_2, x_1) &= (w_\alpha(A - \alpha) + w_\beta(\beta - A))(x_1 - x_2) - (w_\beta - w_\alpha)\mathbb{E}_Y \mathbb{1}\{x_2 \leq Y < x_1\}(x_1 - Y) \\ &= (w_\alpha(0 - \alpha) + w_\beta(\beta - 0))(x_1 - x_2) - (w_\beta - w_\alpha)(x_1 - x_2) \\ &= (w_\beta - w_\alpha)(x_1 - x_2) > 0. \end{aligned}$$

Therefore, post-sorting increases the loss. Next, consider the case where $w_\alpha > w_\beta$ and $\beta w_\beta < \alpha w_\alpha$. Let $Y \equiv x_1$ almost surely. Similar calculation shows that post-sorting increases the loss. $\square$

We prove another subclass of proper loss functions where post-sorting does not increase the loss. From Theorem 4.1 and Theorem 4.2, a proper loss for multiple quantile levels can be written as the following form under mild regularity conditions:

$$\ell((q_n)_{n=1}^N) = \mathbb{E}_Y \sum_{n=1}^N (\mathbb{1}\{Y < q_n\} - \tau_n)(g_n(q_n) - g_n(Y)), \quad (\text{C.19})$$

where g_n is a non-decreasing left-continuous function, for $n \in [N]$.

Proposition C.6. *If a loss function ℓ defined as (C.19) satisfies $g_n(x) = g(x)$ for all $i \in [N]$, then post-sorting does not increase the loss.*

Proof. Due to additivity of the loss function, it suffices to show $\ell(x_1, x_2, y) - \ell(x_2, x_1, y) \geq 0$ for $x_1 > x_2$ where x_n corresponds to the τ_n -level quantile estimate.

$$\begin{aligned} & \ell(x_1, x_2) - \ell(x_2, x_1) \\ &= \mathbb{E}_Y[(\mathbb{1}\{Y < x_1\} - \tau_1)(g_1(x_1) - g_1(Y)) - (\mathbb{1}\{Y < x_2\} - \tau_1)(g_1(x_2) - g_1(Y)) + \\ & \quad (\mathbb{1}\{Y < x_2\} - \tau_2)(g_2(x_2) - g_2(Y)) - (\mathbb{1}\{Y < x_1\} - \tau_2)(g_2(x_1) - g_2(Y))] \\ &= \mathbb{E}_Y[\tau_1(g_1(x_2) - g_1(x_1)) + \mathbb{1}\{Y < x_1\}g_1(x_1) - \mathbb{1}\{Y < x_2\}g_1(x_2) - \mathbb{1}\{x_2 \leq Y < x_1\}g_1(Y) + \\ & \quad \tau_2(g_2(x_1) - g_2(x_2)) + \mathbb{1}\{Y < x_2\}g_2(x_2) - \mathbb{1}\{Y < x_1\}g_2(x_1) + \mathbb{1}\{x_2 \leq Y < x_1\}g_2(Y)] \end{aligned}$$

Now using the fact that $g_1 \equiv g_2 \equiv g$, most of the terms cancel out and we have

$$\ell(x_1, x_2) - \ell(x_2, x_1) = (\tau_2 - \tau_1)(g(x_1) - g(x_2)) \geq 0.$$

□

C.6.3 Omniprediction for multi-horizon forecast

We note that our omniprediction framework can be extended to multi-horizon prediction task, where the ground truth Y_t and base forecasters' predictions $f_b^n(t)$ are not scalars but vectors. By denoting such vector ground truth and vector predictions as $Y_t = (Y_{t,1}, \dots, Y_{t,H})$ and $f_b^n(t) = (f_b^n(t, 1), \dots, f_b^n(t, H))$, respectively, we can define the omniprediction objective for multiple-horizon prediction as

$$\sup_{\ell \in \mathcal{L}_1, b \in [B], h \in [H]} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{q_{n,h} \sim P_{n,h}(t)} [\ell((q_{n,h})_{n=1}^N, Y_{t,h})] - \mathbb{E}[\ell((f_b^n(t, h))_{n=1}^N, Y_{t,h})],$$

which means we want to minimize the maximum of regret for each horizon rather than the average of regret over all horizons, which is commonly used in practice. Then, with Algorithm 1 in hand, the task reduces into solving the following minimax program:

$$\min_{\substack{P_{n,h} \in \Delta_m, \\ \forall n,h}} \max_{\substack{P_{y,h} \in \Delta_m, \\ \forall h}} \sum_{h=1}^H \sum_{n=1}^N \sum_{b=1}^B w_{ibh}^n \mathbb{E}_{q_{n,h} \sim P_{n,h}, Y_h \sim P_{y,h}} [\ell_{\tau_n, \theta_i}(q_{n,h}, Y_h) - \ell_{\tau_n, \theta_i}(f_b^n(t, h), Y_h)].$$

However, this minimax program is of form

$$\min_{\substack{P_{n,h} \in \Delta_m, \\ \forall n,h}} \max_{\substack{P_{y,h} \in \Delta_m, \\ \forall h}} \sum_{h=1}^H g_h((P_{n,h})_{n=1}^N, P_{y,h}) = \sum_{h=1}^H \min_{\substack{P_{n,h} \in \Delta_m, \\ \forall n}} \max_{P_{y,h} \in \Delta_m} g_h((P_{n,h})_{n=1}^N, P_{y,h}).$$

and this reduces into solving H minimax program of form (4.9), solution of which is provided in Theorem 4.3, separately for each horizon. Therefore, running Algorithm 1 for each horizon separately, and combining the solutions together will solve the omniprediction problem for multiple-horizon prediction.

C.6.4 Regret bound of Hedge algorithm

We introduce the well-known regret bound of the Hedge algorithm, which is used frequently applied in the proofs in Chapter 4.

Theorem C.4 (Regret of Hedge (e.g., Theorem 1.5 of [Haz19])). *Consider an online learning problem with B experts receiving bounded losses $\{\ell_b(t)\}_{b \in [B], t \in [T]}$.*

Suppose that at time step t we make the same prediction as expert i with probability

$$w_b(t) = \frac{\exp(-\eta \sum_{s < t} \ell_b(s))}{\sum_{b'=1}^B \exp(-\eta \sum_{s < t} \ell_{b'}(s))},$$

for some $\eta > 0$. Then,

$$\sum_{t=1}^T \sum_{b=1}^B w_b(t) \ell_b(t) \leq \min_{1 \leq b \leq B} \sum_{t=1}^T \ell_b(t) + \eta \sum_{t=1}^T \sum_{b=1}^B w_b(t) \ell_b^2(t) + \frac{\log B}{\eta}.$$