Model Ensembling for Constrained Optimization

Ira Globus Harris
University of Pennsylvania, Philadelphia, PA, USA

Varun Gupta
University of Pennsylvania, Philadelphia, PA, USA

Michael Kearns
University of Pennsylvania, Philadelphia, PA, USA

Aaron Roth
University of Pennsylvania, Philadelphia, PA, USA

Abstract

Many instances of decision making under objective uncertainty can be decomposed into two steps: predicting the objective function and then optimizing for the best feasible action under the estimate of the objective vector. We study the problem of ensembling models for optimization of uncertain linear objectives under arbitrary constraints. We imagine we are given a collection of predictive models mapping a feature space to multi-dimensional real-valued predictions, which form the coefficients of a linear objective that we would like to optimize. We give two ensembling methods that can provably result in transparent decisions that strictly improve on all initial policies. The first method operates in the "white box" setting in which we have access to the underlying prediction models and the second in the "black box" setting in which we only have access to the induced decisions (in the downstream optimization problem) of the constituent models, but not their underlying point predictions. They are *transparent* or *trustworthy* in the sense that the user can reliably predict long-term ensemble rewards even if the instance by instance predictions are imperfect.

2012 ACM Subject Classification Computing methodologies → Learning settings

Keywords and phrases model ensembling, trustworthy AI, decision-making under uncertainty

Digital Object Identifier 10.4230/LIPIcs.FORC.2025.14

1 Introduction

Many instances of decision making under uncertainty can be decomposed into two steps: *prediction* and *optimization*. For example, when deciding on a portfolio of investment assets, we might first predict the returns of individual assets, then choose the portfolio that maximizes predicted return subject to budget and risk constraints. Similarly, when deciding on a route to drive, we might first predict the congestion along each road segment, and then solve a shortest-path problem on the road network to minimize predicted travel time. The predictive component of such a decision making algorithm would take as input a context relevant to the task at hand (e.g. past returns, weather conditions, time of day, etc.) and would produce a vector-valued prediction (e.g. the return of each stock, or the congestion of each road). When paired with a corresponding optimization problem (e.g. maximizing returns subject to risk constraints or minimizing travel time) the predictive component induces a policy, mapping contexts to feasible actions. In practice, such a model of decision making is useful in a variety of settings, from healthcare and delivery services to resource scheduling and inventory stock allocation (see e.g. [7, 20, 1, 2, 3, 6]). Often, in practice, the predictive component is used to estimate the demand for some scarce resource (e.g. medical diagnostic tests, rental vehicles, retail inventory) and then the optimization is used to make allocation decisions satisfying real-world operational constraints (e.g. storage space, production cost, transportation time, personnel capacity).

Now suppose we have multiple such predictive models that make (different) predictions, and produce (different) "policies" – informally, the recommended decision for any particular context. In this paper, we develop methods for *ensembling* these systems to produce policies that 1) provide *transparent* decisions in that they guarantee that the actual expected reward of the ensemble is equal to its predicted reward and 2) can strictly improve on the constituent or base policies. Our second goal is similar in motivation to classical ensembling methods such as boosting and bagging (see e.g. [19]), but in a much more complex action space in which contexts are mapped to high dimensional actions via constrained optimization – and we are able to accomplish this with transparent methods. We give two such ensembling methods. The first operates in a *white box* model, in which we have direct access to the constituent predictive models, and not just to the policies they induce. The second operates in a *black box* model, in which we do not have access to the constituent models' point predictions but only to their induced policies, and assume nothing about how they are derived – e.g. they may or may not be the result of optimizing over predictions.

Our transparency guarantees, in the vein of those from [16], can be thought of as a trustworthiness condition on the decisions made by the ensemble. Both proposed methods have the advantage that, after refining the constituent models' policies, the resulting ensemble's evaluations of its own decisions are approximately accurate on average – neither substantially over- or under-confident about the realized outcomes – conditioned on the decisions it itself made. With a system satisfying this condition, a decision maker can then trust that following the ensemble's recommendations will result in the promised outcomes. This is alone a useful condition to provide on top of the base models in our setting, and we are able to do so while maintaining usefulness of the base models' decisions.

1.1 Our Results

We study a setting in which a decision maker solves a d-dimensional optimization problem with an objective that is a linear function of the d label variables, denoted y. The decision maker does not know the objective, but instead has estimates of the d label variables from a collection of k predictive models. The central question is: what action should the decision maker take, given (differing) predictions and recommended actions from the collection of models? The optimization problem can be defined by a set of specified (and not necessarily convex) constraints.

Our white box ensembling method follows a simple, intuitive idea. Given a context vector x, each of the k constituent models, denoted h_i for $i \in [k]$, produces a *predicted* label vector $h_i(x)$, which, after solving the corresponding optimization problem produces the action $\pi_i(x)$. If model i's predictions were correct, then the corresponding payoff of the model's action would be $\pi_i(x) \cdot h_i(x)$. We call this model i's self-assessed payoff. The idea is to "transparently ensemble" the models by always taking the action of the model with the largest self-assessment: $\arg\max_i \pi_i(x) \cdot h_i(x)$. But this idea runs into several obstacles. First, the self-assessed expected payoff of a model $\mathbb{E}[\pi_i(x) \cdot h_i(x)]$ need not have any clean relationship with its actual expected payoff $\mathbb{E}[\pi_i(x) \cdot y]$. Second, even if each model is "self-consistent" in the sense that $\mathbb{E}[\pi_i(x) \cdot y] = \mathbb{E}[\pi_i(x) \cdot h_i(x)]$, we would expect to lose self-consistency after selecting the model with the highest self-assessment, because we would be conditioning on a model having an unusually high self-assessed payoff: this would result in upward bias for the selected model *conditional on the selection event* even if the models produced independent, unbiased forecasts of the outcome variables. We solve these problems by showing how to efficiently post-process the constituent models so that they have consistent self-assessments

even conditional on their selection – using techniques from the multicalibration literature [14]. For each model i in the set that we are ensembling, our conditions guarantee:

$$\mathbb{E}\left[\pi_i(x) \cdot (h_i(x) - y) \middle| i = \arg\max_{i'} \pi_{i'}(x) \cdot h_{i'}(x)\right] = 0.$$

We show that the ensembled policy π^* that results from selecting the action $\pi^*(x) = \pi_{i^*(x)}(x)$ (where $i^*(x) = \arg\max_i[\pi_i(x) \cdot h_i(x)]$ is the index of the model with the highest self-assessment) is self-consistent and is guaranteed to obtain expected payoff that is at least (up to error terms):

$$\begin{aligned}\mathbb{E}\left[\pi^*(x) \cdot y\right] &= \mathbb{E}\left[\pi^*(x) \cdot h_{i^*}(x)\right] \\ &\geq \mathbb{E}\left[\max_i \pi_i(x) \cdot h_i(x)\right].\end{aligned}$$

Note that the maximum is inside the expectation, and so this is the *point-wise* maximum self-assessed payoff. This can be substantially higher than the expected payoff of the best constituent model, which is (because of the self-consistency condition) $\max_i \mathbb{E}[\pi_i(x) \cdot y] = \max_i \mathbb{E}[\pi_i(x) \cdot h_i(x)]$.

Our white box ensembling method offers a strong guarantee, but has three limitations. First, it requires access to the predictive model $h_i(x)$ that underlies the policy $\pi_i(x)$ This might not always be available – for example, if the decision maker receives action suggestions from a "prediction as a service" provider or a recommendation system, which may not be based on underlying predictions of the objective coefficients, or even if so, may not be made visible to the decision maker. Second, it requires updating and maintaining all of the constituent models to be ensembled, which might be prohibitive if the number of models is large.

Towards addressing these limitations, we introduce an alternative "black box" ensembling method. This method maintains only a single predictive model, and only requires access to the collection of k policies $\pi_i(x)$ to be ensembled, not any details of their implementation (i.e. the constituent models or their point predictions). In broad strokes, it works by maintaining a single predictive model $h^*(x)$ for the labels that is unbiased both conditional on each coordinate of its own induced action $\pi^*(x)$ and conditional on each coordinate of the actions $\pi_i(x)$ chosen by each model to be ensembled. We prove a swap regret-like guarantee for this ensemble: not only is it self-consistent, but it obtains higher payoff than any of the constituent models even conditional on any coordinate of its induced action. Because this technique only requires maintaining a single predictive model and requires fewer evaluations of the downstream optimization problem, it can be trained substantially faster than our white box ensembling method. However, as a consequence of the limitations of its black box access to the constituent policies, it does not give the same form of strong point-wise guarantee that the white box approach does.

1.2 Related Work

The problem of finding policies that solve optimization problems in the face of unknown label vectors is often solved by first predicting the label vectors and then optimizing for the predicted label. The "Smart Predict then Optimize" framework of [5] focuses on the design of surrogate loss functions to minimize in the prediction training phase that are best suited for the particular downstream optimization task. Our work differs from this in that we do not train a single model using a surrogate loss function that incorporates the downstream optimization, but instead leverage techniques from the multicalibration literature [14] to ensemble a collection of models to provide transparent and utility-preserving guarantees on the downstream task.

A line of work on omniprediction in both the unconstrained and constrained settings [12, 8, 15, 10, 13] gives theorems that informally state that if a 1-dimensional predictor f (usually for a binary outcome) is multicalibrated with respect to a set of models, then for some family of loss functions, f has loss at most the loss of the best model in the class. Here the focus is generally on the ability of the model f to perform well over multiple loss functions, and the promise is only that f performs as well as the best model in the class, rather than strictly better than it. One notable exception is [11] which analyzes multicalibration as a boosting algorithm for regression functions, and proves that if f is multicalibrated with respect to a class of models $\mathcal{H}$, then it in fact performs as well as the best model in a strictly more expressive class (that $\mathcal{H}$ serves as weak learners for). Similarly, [9] and [18] consider the problem of reconciling two 1-dimensional regression functions that have similar error, but make different predictions. They show how to combine two such models into a single, more accurate model. A primary point of departure for us is that we consider ensembling models for high dimensional optimization problems, rather than 1-dimensional classification and regression problems.

The debiasing steps that we use are closest in spirit to those used in "decision calibration" [21] or "prediction for action" [16], which aim to produce predictors that are unbiased conditional on the action taken by some downstream decision maker. Independently and concurrently of our work, [4] adapt the "reconciliation" procedure of [18] to the decision calibration setting, updating pairs of models that frequently induce different decisions in downstream decision makers on the regions on which they induce different downstream decisions. Their end result is two new and reconciled models which agree with one another and are unbiased conditional on the action induced – and their bounds inherit a polynomial dependence on k, the number of actions of the downstream decision maker. Because the optimization problems we consider have linear objectives, we only need that our predictions are unbiased subject to the *coordinates* of the actions that result from optimization – a fact that was also used by [16]. This is what lets us handle downstream optimization problems with very large action spaces. The focus of [21, 16] was on producing policies that offer a downstream decision maker various forms of low regret guarantees – in contrast, our interest is in ensembling multiple explicit policies. The focus of [4] is on reconciling model multiplicity, whereas our focus is on achieving superior performance for the task at hand than the base models, and not explicitly on resolving disagreements between them.

2 Preliminaries

We assume there is a joint probability distribution $\mathcal{D}$ over a *context space* $\mathcal{X}$ and a d-dimensional real-valued label space $\mathcal{Y} \subset \mathbb{R}^d$. Label vectors $y \in \mathcal{Y}$ are assumed to have bounded coordinates: $\|y\|_\infty \leq M$ for all $y \in \mathcal{Y}$.

There is an underlying optimization problem, to map contexts x to d-dimensional actions $a \in \Omega \subseteq [0,1]^d$. Here Ω represents a known but arbitrary feasibility constraint – there is, e.g., no requirement that Ω be convex. The payoff of an action $a \in \Omega$ given a label vector $y \in \mathcal{Y}$ is modeled as their inner product $a \cdot y$.

If the labels y were known at each round, then the optimal action to take would be $\arg\max_{a \in \Omega} a \cdot y$ – the solution to an optimization problem with a linear objective and arbitrary constraints represented by Ω. However, the labels y are not known. One way to approach the decision-making under uncertainty problem is to first train a predictive model $h : \mathcal{X} \to \mathcal{Y}$ that maps contexts to predicted label vectors. We suppose that the decision maker has k such models and then must decide on what action to take for each context.

▶ **Definition 1** (Policy). *A policy π is a mapping from contexts to feasible actions $\pi : \mathcal{X} \to \Omega$.*

The payoff of a policy π on a specific context x given a label vector y is the inner product of the actions that π induces on x and the vector y: $\pi(x) \cdot y$. The expected payoff of a policy π is the $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\pi(x) \cdot y]$.

Each predictive model h *induces* a policy that finds the action that maximizes *predicted* payoff given the constraints.

▶ **Definition 2** (Model Induced Policy). *Fix a model $h : \mathcal{X} \to \mathcal{Y}$. We say that model h induces a policy π_h, defined as*

$$\pi_h(x) \coloneqq \arg\max_{a\in\Omega} a \cdot h(x),$$

for each $x \in \mathcal{X}$.

A model h induces a policy π_h that has *actual* expected payoff $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\pi_h(x) \cdot y]$. We will also be interested in a model's self-assessed or self-evaluated payoff: given an example x, a model h's self-assessed payoff is $\pi_h(x) \cdot h(x)$, and a model's expected self-assessment is $\mathbb{E}[\pi_h(x) \cdot h(x)]$. Absent further conditions, a model's self-assessed payoff need not have any relationship to its actual payoff. In the next section, we will define relevant conditions that we will impose on a model to relate its self-assessed payoff with its actual payoff.

These conditions reference a partition, or bucketing, of the unit interval $[0, 1]$ into $\frac{1}{w}$ level sets each of width w that we will refer to as $\mathcal{T}$. We refer to specific level sets in $\mathcal{T}$ as τ and the midpoint of an interval τ as τ_1.

2.1 Consistency Conditions

In order to relate a model's self-assessed payoff to its actual payoff, we will leverage the ability to make conditionally "consistent" predictions – informally, predictions which are accurate on average, marginally, but also conditional on arbitrary sets of interest. The parameterization we choose is related to the multicalibration literature – see e.g. [17] for a discussion of this and alternative parameterizations.

▶ **Definition 3** (Consistent Predictions). *Fix a distribution $\mathcal{D}$. We say that the model $h : \mathcal{X} \to \mathcal{Y}$ is $\alpha-$consistent with respect to a collection of sets $\mathcal{C} \subseteq 2^{\mathcal{X}}$ if $\forall\ C \in \mathcal{C}$, it is the case that*

$$\|\mathbb{E}_{\mathcal{D}}[y - h(x)|x \in C]\|_\infty \leq \frac{\alpha}{\Pr[x \in C]}.$$

One important class of sets we will be concerned with making consistent predictions on is the level sets of different policies we aim to ensemble.

▶ **Definition 4** (Policy Level Sets). *Fix a policy π and bucketing $\mathcal{T}$. We refer to the level sets of a policy π as*

$$\mathcal{C}_\pi^{\mathcal{T}} = \{\{x \mid \pi(x)_i \in \tau\} : i \in [d], \tau \in \mathcal{T}\},$$

which is the collection of subsets of $\mathcal{X}$ on which π induces an action that allocates an amount in the interval τ to outcome coordinate i.

For shorthand, if a model is consistent with respect to a collection of sets $\mathcal{C}_\pi^{\mathcal{T}}$ for some policy π, we say that it is consistent to that policy.

▶ **Definition 5** (Consistency to a Policy). *Fix a model $h : \mathcal{X} \to \mathcal{Y}$, policy $\pi : \mathcal{X} \to \Omega$, and bucketing $\mathcal{T}$. We say that the model h is $\alpha-$consistent with respect to the policy π if it is $\alpha-$consistent with respect to the level sets of π. That is, if for all $i \in [d], \tau \in \mathcal{T}$, it is the case that*

$$\|\mathbb{E}_{\mathcal{D}}[y - h(x)|\pi(x)_i \in \tau]\|_\infty \leq \frac{\alpha}{\Pr[\pi(x)_i \in \tau]}.$$

As we will see, we will be able to post-process a model h to be consistent with collections of sets $\mathcal{C}$ that may be defined in terms of h itself. In particular, it will be useful for us to ask that a model h's predictions are consistent with respect to the policy π_h that it itself induces. This will turn out to be the condition we need to make a model "self-consistent" – the transparency condition in the sense that its self-assessed payoff accords with its actual payoff. Thus when a model h is consistent (in the sense of Definition 3) with π_h, we will say that h is *self-consistent*.

▶ **Definition 6** (Self-Consistent Model). *Fix a model $h : \mathcal{X} \to \mathcal{Y}$ and bucketing $\mathcal{T}$. We say that the model h is $\alpha-$self-consistent if it is $\alpha-$consistent with respect to the level sets of π_h, where π_h is the policy induced by the model h (Definition 2). That is, if for all $i \in [d], \tau \in \mathcal{T}$, it is the case that*

$$\|\mathbb{E}_{\mathcal{D}}[y - h(x)|\pi_h(x)_i \in \tau]\|_\infty \leq \frac{\alpha}{\Pr[\pi_h(x)_i \in \tau]}.$$

3 Consistent Predictions

In this section, we describe the procedure for making consistent predictions and the transparent outcome guarantees that consistent models provide. The basic algorithm driving our approach is an iterative de-biasing procedure similar to the template that has become common in the multi-calibration literature [14]. At a high level it iteratively identifies subsets of the data domain on which the current model exhibits statistical bias. It then shifts the model's predictions on these identified regions to remove the statistical bias. Where we will differ from the multicalibration literature will be in our choice of bias events – these will turn out to be "cross-calibration" events defined across multiple models, and defined in terms of the solution to the optimization problem induced by the prediction of our models. For simplicity we describe the algorithm as if it has access to the distibution $\mathcal{D}$ directly, but out-of-sample guarantees follow straightforwardly from standard techniques [17].

Algorithm 1 Update (hyperparameters: consistency tolerance α).

Input model h^0, collection of sets $\mathcal{C} \subseteq 2^{\mathcal{X}}$
Initialize $t = 0$
while $\exists\, C \in \mathcal{C}$ such that $\Pr[x \in C]\|\mathbb{E}_{\mathcal{D}}[y - h^t(x)|x \in C]\|_\infty > \alpha$ **do**
  $t = t + 1$
  $\vec{\delta} \coloneqq \mathbb{E}_{\mathcal{D}}[y - h^{t-1}(x)|x \in C]$
  $h^t(x) \coloneqq h^{t-1}(x) + \mathbb{1}_{[x \in C]} \cdot \vec{\delta}$
end while
Output model h^t

The convergence analysis of this algorithm follows from the fact that correcting statistical bias on a subset of the data domain is guaranteed to decrease the squared error of a model. Thus squared error can act as a potential function, even when the subsets on which we update the model intersect. The following is a standard lemma, first used by [14] in the multicalibration literature – we adopt a variant used in [17].

▶ **Lemma 7** (Monotone Decrease of Squared Error [17]). *Fix a model h, distribution $\mathcal{D}$, policy π, and set $C \subseteq \mathcal{X}$. Let*

$$\Delta = \mathbb{E}_{\mathcal{D}}[y|x \in C] - \mathbb{E}_{\mathcal{D}}[h(x)|x \in C]$$

and

$$h'(x) = \begin{cases} h(x) + \Delta & \text{if } x \in C, \\ h(x) & \text{o.w.} \end{cases}$$

Then,

$$\mathbb{E}_{\mathcal{D}}[\|h(x) - y\|_2^2] - \mathbb{E}_{\mathcal{D}}[\|h'(x) - y\|_2^2] = \Pr[x \in C] \cdot \|\Delta\|_2^2.$$

The convergence of the algorithm then follows from a potential argument.

▶ **Lemma 8.** *The procedure* Update$(h, \mathcal{C})$ *(Algorithm 1) terminates within dM^2/α^2 rounds.*

All proofs will be deferred to Appendix A.

3.1 Using Consistent Predictions

In this section, we show that consistency with respect to carefully constructed sets of events allows us to evaluate the payoff of a policy induced by a model via its *self-assessments*, and lets us compare the policy induced by a model with other policies. These statements will be the basic building blocks of our ensembling methods.

First we show that if a model h is consistent with respect to a policy π's level sets (Definition 4), then the model h's predicted label can be used in place of the true label y to correctly estimate the payoff of π.

▶ **Lemma 9.** *Fix a distribution $\mathcal{D}$ and a bucketing $\mathcal{T}$, with $w = \sqrt{\alpha/M}$. Let $\pi : \mathcal{X} \to \Omega$ be an arbitrary policy. Let $h : \mathcal{X} \to \mathcal{Y}$ be a model that is $\alpha-$consistent with respect to π (Definition 5). Then: $|\mathbb{E}_{\mathcal{D}}[\pi(x) \cdot h(x)] - \mathbb{E}_{\mathcal{D}}[\pi(x) \cdot y]| \leq 2d\sqrt{\alpha M}$.*

An especially useful special case of Lemma 9 is the case in which a model h is consistent with the policy π_h that it itself induces. In this case, the model can be used to correctly evaluate its own payoff (up to error terms), and corresponds to a natural and useful "transparency" condition similar in spirit to the transparency conditions of [21, 16].

▶ **Corollary 10.** *Fix a distribution $\mathcal{D}$ and a bucketing $\mathcal{T}$, with $w = \sqrt{\alpha/M}$. If a model $h : \mathcal{X} \to \mathcal{Y}$ is $\alpha-$self-consistent, then its expected outcome is close to its expected self-evaluation:*

$$|\mathbb{E}_{\mathcal{D}}[\pi_h(x) \cdot h(x)] - \mathbb{E}_{\mathcal{D}}[\pi_h(x) \cdot y]| \leq 2d\sqrt{\alpha M}.$$

Comparing Policies

It is also possible for a predictor to satisfy the consistency conditions with respect to *multiple* policies. Why might this be useful? Informally, since you can trust a model's evaluation of any policy that it is consistent with respect to, if a model is consistent with many policies, optimizing according to its predictions should induce outcomes that are only better than those from the best policy it is consistent with. The following shows that this is in fact the case.

▶ **Lemma 11.** *Fix a distribution $\mathcal{D}$ and a bucketing $\mathcal{T}$, with $w = \sqrt{\alpha/M}$. Let $\pi : \mathcal{X} \to \Omega$ be an arbitrary policy. Let $h : \mathcal{X} \to \mathcal{Y}$ be an $\alpha-$self-consistent model that is also $\alpha-$consistent with respect to the policy π. Then: $\mathbb{E}_{\mathcal{D}}[\pi_h(x) \cdot y] \geq \mathbb{E}_{\mathcal{D}}[\pi(x) \cdot y] - 4d\sqrt{\alpha M}$.*

4 White Box Ensembling Method

This section describes the first of two ensembling methods: a "white box" method for ensembling k models. Like the method we will describe in Section 5, this method enjoys transparent outcome guarantees. However, this method requires strong access to the models being ensembled – access to their point predictions, rather than just the policies induced by the model. We prove that the final ensembled policy strictly improves on the reward of the constituent models after debiasing, by obtaining their *pointwise maximum* self-assessed reward. The debiasing procedure is guaranteed to improve the squared error of the constituent predictive models, but not necessarily the reward of the policies they induce.

Interaction Model

A decision maker has access to k constituent models makings predictions of the coefficients of a linear objective function (e.g. stock prices), which they are using to make decisions subject to some arbitrary constraints. They build an ensemble using these k models by updating them to satisfy consistency conditions which we describe formally below. In this scheme, the decision maker needs access not only to the policies they are incorporating into their decision making procedure, but the predictive models used to induce these policies, as the ensembling procedure involves iteratively modifying the predictions of each of these constituent models.

▶ **Definition 12** (White Box Ensemble). *A white box ensemble* $\mathbf{h} = h_1, \ldots, h_k$ *is a collection of* k *models.*

The white box ensemble policy that the decision maker will employ is simple: selecting the constituent model that has the highest self-assessed payoff (or lowest, if the downstream optimization is a minimization). For simplicity, we will refer to the index $i^*(x)$ as the model in a white box ensemble that has the highest self-assessed payoff on a given point x: $i^*(x) = \arg\max_{j \in [k]} \pi_{h_j}(x) \cdot h_j(x)$.

▶ **Definition 13** (White Box Ensemble Policy). *A white box ensemble policy* $\pi_{\mathbf{h}}$ *for a white box ensemble* $\mathbf{h} = (h_1, \ldots, h_k)$ *of models is the policy that, for each* $x \in \mathcal{X}$*, outputs* $\pi_{h_{i^*(x)}}(x)$*.*

The expected return of a white box ensemble $\mathbf{h}$ is $\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x) \cdot y] = \sum_{i \in [k]} \mathbb{E}_{\mathcal{D}}[\mathbb{1}_{i=i^*(x)} \cdot \pi_{h_i}(x) \cdot y]$. The expected self-evaluation of a white box ensemble is $\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x) \cdot \mathbf{h}(x)] = \sum_{i \in [k]} \mathbb{E}_{\mathcal{D}}[\mathbb{1}_{i=i^*(x)} \cdot \pi_{h_i}(x) \cdot h_i(x)]$.

4.1 Ensembling Models

In this method, we will require consistency of our predictions with respect to another special collection of conditioning events: the sets on which each constituent model is most "optimistic" – or has the highest self-assessed payoff.

▶ **Definition 14** (Maximum Model Level Sets). *Fix a set of* k *models* $h_i^t : \mathcal{X} \to \mathcal{Y}$ *for* $i \in [k]$*. We refer to the maximum model level sets as*

$$\mathcal{C}^t_{[k]} = \Bigg\{ \big\{ x | h_i^t(x) \cdot \pi_{h_i^t}(x) \geq h_j^t(x) \cdot \pi_{h_j^t}(x) \quad \forall\ j \in [k] \big\} : i \in [k] \Bigg\}.$$

These correspond to the sets of examples on which each of the models i *has the highest self-assessed payoff.*

The procedure for ensembling k models involves modifying the constituent models so each is consistent with respect to the level sets of its own induced policy conditional on the identity of the model with the highest self-evaluated payoff. Informally, consistency on this set of events is useful because they are related to how the decision maker takes actions – the ensemble follows the action of policy i exactly when it has the highest self-assessed payoff. If each constituent model's predictions are self-consistent and consistent conditional on these sets characterizing when the decision maker is taking different actions, the resulting ensembling satisfies strong outcome guarantees.

Algorithm 2 White Box Ensembling.

Select consistency tolerance α and discretization $\mathcal{T}$
Initialize $t = 0$
Input Initial models $h_i^t, i \in [k]$
while $\exists\ i$ s.t. $\exists\ C \in \mathcal{C}^{\mathcal{T}}_{\pi_{h_i^t}} \times \mathcal{C}^t_{[k]}$ s.t. $\Pr[x \in C] \|\mathbb{E}_{\mathcal{D}}[y - h_i^t(x) | x \in C]\|_\infty > \alpha$ **do**
  $h_i^{t+1} = \textsc{Update}(h_i^t, \mathcal{C}^{\mathcal{T}}_{\pi_{h_i^t}} \times \mathcal{C}^t_{[k]})$, for each $i \in [k]$
  $t = t + 1$
end while
Output white box ensemble $\mathbf{h} = (h_1^t, \ldots, h_k^t)$

4.2 Convergence of White Box Ensembling

Convergence of the white box ensembling method follows similarly to convergence of Algorithm 1. Algorithm 2 repeatedly calls Algorithm 1 as a subroutine on each of the k constituent models, on an adaptively chosen sequence of conditioning events.

▶ **Lemma 15.** *Fix a model $h^1 : \mathcal{X} \to \mathcal{Y}$ and consistency tolerance α. Let $h^t = \textsc{Update}(h^{t-1}, \mathcal{C}^{t-1})$ for $t > 1$. For any sequence of, possibly adaptive, conditioning events $\mathcal{C}^1, \ldots, \mathcal{C}^t$, this process will terminate after at most $\frac{dM^2}{\alpha^2}$ rounds: that is, for $t > \frac{dM^2}{\alpha^2}$, it is the case that $h^t = \textsc{Update}(h^{t-1}, \mathcal{C}^{t-1})$.*

▶ **Lemma 16.** *Algorithm 2 terminates in $k \cdot \frac{dM^2}{\alpha^2}$ rounds.*

4.3 Utility Guarantees

We now analyze our white box ensembling method. First, we prove that it is self-consistent – its self-assessed payoff is equal (up to error terms) to its realized payoff, in expectation.

▶ **Lemma 17.** *Fix distribution $\mathcal{D}$ and bucketing $\mathcal{T}$, with $w = \sqrt{\alpha k / M}$. Let $\mathbf{h}$ be an ensemble of k models in which each h_i is $\alpha-$consistent with respect to $\mathcal{C}^{\mathcal{T}}_{\pi_{h_i}} \times \mathcal{C}_{[k]}$ for $i \in [k]$. The expected self-evaluation of the ensemble $\mathbf{h}$ is approximately equal to its expected revenue: $\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x) \cdot y] \geq \mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x) \cdot \mathbf{h}(x)] - 2d\sqrt{\alpha k M}$.*

We next prove two bounds on the performance of the method. The first states that – up to error terms – the payoff of the ensemble is equal to the *expected maximum* self assessed payoff of each of the constituent models. Notice that here we bound the performance of the ensemble by the expected max, which is larger than the max expectation, the latter of which corresponds to the best single constituent model.

▶ **Lemma 18.** *Fix distribution $\mathcal{D}$ and bucketing $\mathcal{T}$, with $w = \sqrt{\alpha k/M}$. Let $\mathbf{h}$ be an ensemble of k models in which h_i is $\alpha-$consistent with respect to $\mathcal{C}^{\mathcal{T}}_{\pi_{h_i}} \times \mathcal{C}_{[k]}$ for $i \in [k]$. Then,*

$$\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x) \cdot y] \geq \mathbb{E}_{\mathcal{D}}[\max_{j \in [k]} \pi_j(x) \cdot h_j(x)] - 2d\sqrt{\alpha k M}.$$

The next performance bound is a "swap-regret" like guarantee. It states that on the subset of examples on which the model chooses to follow policy i, it could not have improved by instead following some other policy j – simultaneously for all i and j.

▶ **Lemma 19.** *Fix distribution $\mathcal{D}$ and bucketing $\mathcal{T}$, with $w = \sqrt{\alpha k/M}$. Let $\mathbf{h}$ be an ensemble of k models in which h_i is $\alpha-$consistent with respect to $\mathcal{C}^{\mathcal{T}}_{\pi_{h_i}} \times \mathcal{C}_{[k]}$ for $i \in [k]$. Then, for all $\phi : [k] \to [k]$,*

$$\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x) \cdot y] \geq \mathbb{E}_{\mathcal{D}}[\pi_{h_{\phi(i^*(x))}}(x) \cdot y] - 4d\sqrt{\alpha k M}.$$

5 Black Box Ensembling Method

In this section, we describe the second, "black box," method to ensemble models. This method maintains a single, deterministic predictor which can be easily updated in the presence of new information, and requires only access to the induced policy of the models being ensembled. Like the method described in Section 4, the black box ensembling method enjoys transparent outcome guarantees. We show that this ensembling provides a "swap style" utility guarantee, that the ensemble provably induces a payoff as high as any of the policies it is consistent to, conditioned on its action.

Interaction Model

A decision maker has a predictive model h and access to k arbitrary policies $\pi_1, \ldots, \pi_k$, whose information they want to incorporate into their predictive model. The decision maker builds an ensemble by updating their model to satisfy consistency conditions relating to each policy π_i which we describe below. Since this procedure only involves updating a single predictive model, the one that the decision maker begins with, they are able to ensemble policies generated arbitrarily - e.g. even ones without an underlying predictive model.

5.1 Ensembling Policies

This method maintains a model that is unbiased with respect to the policy it itself induces, as well as each constituent policy to be ensembled.

Algorithm 3 Black Box Ensembling.

Input collection of k policies $\{\pi_1, \ldots, \pi_k\}$
Select consistency tolerance α and discretization $\mathcal{T}$
Initialize $t = 0$ and model h^t
while $\exists\, C \in \mathcal{C}^{\mathcal{T}}_{\pi_{h^t}} \cup_{i \in [k]} \mathcal{C}^{\mathcal{T}}_{\pi_i}$ s.t. $\Pr[x \in C] \|\mathbb{E}_{\mathcal{D}}[y - h^t(x) | x \in C]\|_\infty > \alpha$ **do**
  $h^{t+1} = \textsc{Update}(h^t, \mathcal{C}^{\mathcal{T}}_{\pi_{h^t}} \cup_{i \in [k]} \mathcal{C}^{\mathcal{T}}_{\pi_i})$
  $t = t + 1$
end while
Output model h^t

5.2 Utility Guarantees

We now state and prove our main utility statement for our black box ensembling method: A swap-regret style guarantee that establishes that the policy induced by the ensemble model performs better than any of its constituent policies, not just overall, but also conditional on level sets of any policy in its ensemble.

▶ **Lemma 20.** *Fix distribution $\mathcal{D}$ and bucketing $\mathcal{T}$. Let h be an $\alpha-$self-consistent model that is $\alpha-$consistent with respect to a collection of policies $\mathcal{P}$. Then, $\mathbb{E}_{\mathcal{D}}[\pi_h(x) \cdot y | x \in C] \geq \mathbb{E}_{\mathcal{D}}[\pi(x) \cdot y | x \in C] - \frac{2\alpha d}{\Pr[\pi(x)_i \in \tau]} - 2wMd$, for all $\pi, \pi' \in \mathcal{P} \cup \{\pi_h\}$ and all $C \in \mathcal{C}^{\mathcal{T}}_{\pi'}$.*

6 Conclusion

We introduce two simple ensembling methods for a decision maker to make transparent decisions in a $d-$dimensional optimization problem with an (uncertain) linear objective function and arbitrary constraints, when they are given a collection of k predictive models to estimate the coefficients of the objective. The methods allow for *transparent* decision making, as the decision maker is promised that the average realized payoff of the recommended policy will be close to the self-evaluated (i.e. predicted) payoff of the policy. The first ensembling method operates in a "white box" access model, in which the decision maker has full access to the underlying constituent models (the point predictions). The second operates in a weaker, "black box" access model, in which the decision maker only has access to the induced policies (i.e. the recommended actions) of the constituent models.

6.1 Discussion of Limitations

Our algorithms operate in the batch/distributional setting, and the guarantees we prove are limited to when the data at deployment time is distributed identically to the training data. An important open question is how to adapt similar practical techniques to give robust methods that allow for various kinds of distribution shift. Techniques of [16] could be adapted to give similar ensembling algorithms in the online adversarial setting when the prediction target can be observed shortly after prediction at test time; these algorithms are mainly of theoretical interest, and practical variants would need to be developed. Even in the batch setting, our white box ensembling method is computationally expensive, and more efficient algorithms would be important improvements.

References

1 Hongrui Chu, Wensi Zhang, Pengfei Bai, and Yahong Chen. Data-driven optimization for last-mile delivery. *Complex & Intelligent Systems*, 9(3):2271–2284, 2023.

2 Sarang Deo, Kumar Rajaram, Sandeep Rath, Uday S Karmarkar, and Matthew B Goetz. Planning for hiv screening, testing, and care at the veterans health administration. *Operations research*, 63(2):287–304, 2015. doi:10.1287/OPRE.2015.1353.

3 Priya Donti, Brandon Amos, and J Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 5484–5494, 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/3fc2c60b5782f641f76bcefc39fb2392-Abstract.html.

4 Ally Yalei Du, Dung Daniel Ngo, and Zhiwei Steven Wu. Reconciling model multiplicity for downstream decision making. *CoRR*, 2024. doi:10.48550/arXiv.2405.19667.

5 Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

6 Jérémie Gallien, Adam J Mersereau, Andres Garro, Alberte Dapena Mora, and Martín Nóvoa Vidal. Initial shipment decisions for new products at zara. *Operations Research*, 63(2):269–286, 2015. doi:10.1287/OPRE.2014.1343.

7 Daniele Gammelli, Yihua Wang, Dennis Prak, Filipe Rodrigues, Stefan Minner, and Francisco Camara Pereira. Predictive and prescriptive performance of bike-sharing demand forecasts for inventory management. *Transportation Research Part C: Emerging Technologies*, 138:103571, 2022.

8 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. doi:10.1137/1.9781611977912.98.

9 Sumegha Garg, Michael P Kim, and Omer Reingold. Tracking and improving information in the service of fairness. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 809–824, 2019. doi:10.1145/3328526.3329624.

10 Ira Globus-Harris, Varun Gupta, Christopher Jung, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Multicalibrated regression for downstream fairness. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 259–286, 2023. doi:10.1145/3600211.3604683.

11 Ira Globus-Harris, Declan Harrison, Michael Kearns, Aaron Roth, and Jessica Sorrell. Multicalibration as boosting for regression. In *Proceedings of the 40th International Conference on Machine Learning*, pages 11459–11492, 2023. URL: https://proceedings.mlr.press/v202/globus-harris23a.html.

12 Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2022.

13 Parikshit Gopalan, Princewill Okoroafor, Prasad Raghavendra, Abhishek Shetty, and Mihir Singhal. Omnipredictors for regression and the approximate rank of convex functions. *arXiv preprint arXiv:2401.14645*, 2024. doi:10.48550/arXiv.2401.14645.

14 Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

15 Lunjia Hu, Inbal Rachel Livni Navon, Omer Reingold, and Chutong Yang. Omnipredictors for constrained optimization. In *International Conference on Machine Learning*, pages 13497–13527. PMLR, 2023. URL: https://proceedings.mlr.press/v202/hu23b.html.

16 Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making, 2023. doi:10.48550/arXiv.2310.17651.

17 Aaron Roth. Uncertain: Modern topics in uncertainty estimation, 2022.

18 Aaron Roth, Alexander Tolbert, and Scott Weinstein. Reconciling individual probability forecasts, 2023. arXiv:2209.01687.

19 Robert E Schapire and Yoav Freund. Boosting: Foundations and algorithms. *Kybernetes*, 42(1):164–166, 2013.

20 Akylas Stratigakos, Simon Camal, Andrea Michiorri, and Georges Kariniotakis. Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy. *IEEE Transactions on Power Systems*, 37(6):4696–4708, 2022.

21 Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 34:22313–22324, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/bbc92a647199b832ec90d7cf57074e9e-Abstract.html.

A Proofs

Proof of Lemma 8. Each round within the procedure UPDATE$(\cdot,\cdot)$ finds a consistency violation with respect to the input model h and collection of sets $\mathcal{C}$ of the form: $\Pr[x \in C]\|\mathbb{E}_{\mathcal{D}}[y - h(x)|x \in C]\|_\infty > \alpha$ for some $C \in \mathcal{C}$. By Lemma 7, we know that the decrease of squared error between the policies of two adjacent rounds of UPDATE$(\cdot,\cdot)$ denoted h and h' satisfy:

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}[\|h(x) - y\|_2^2] - \mathbb{E}_{\mathcal{D}}[\|h'(x) - y\|_2^2] \\ &\quad = \Pr[x \in C] \cdot \|\mathbb{E}_{\mathcal{D}}[y - h(x)|x \in C]\|_2^2.\end{aligned}$$

Additionally, we know that:

$$\|\mathbb{E}_{\mathcal{D}}[y - h(x)|x \in C]\|_2^2 \geq \|\mathbb{E}_{\mathcal{D}}[y - h'(x)|x \in C]\|_\infty^2.$$

By the stopping condition of UPDATE$(\cdot,\cdot)$, we know that while the procedure has not terminated:

$$\max_{C \in \mathcal{C}} \Pr[x \in C]\|\mathbb{E}_{\mathcal{D}}[y - h(x)|x \in C]\|_\infty > \alpha.$$

Therefore, we know that in each iteration of UPDATE$(\cdot,\cdot)$ the squared error of the model h drops by

$$\mathbb{E}_{\mathcal{D}}[\|h(x) - y\|_2^2] - \mathbb{E}_{\mathcal{D}}[\|h'(x) - y\|_2^2] \geq \frac{\alpha^2}{\Pr[x \in C]} \geq \alpha^2.$$

Since each $h(x) \in [0, M]^d \; \forall \; x \in \mathcal{X}$ and $\mathcal{Y} \subseteq [0, M]^d$, we know that the squared error can be at most $d \cdot M^2$. ◀

Proof of Lemma 9. First, we show that $\mathbb{E}_{\mathcal{D}}[\pi(x) \cdot y] \geq \mathbb{E}_{\mathcal{D}}[\pi(x) \cdot h(x)] - 2d\sqrt{\alpha M}$.

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}[\pi(x) \cdot y] \\ &\qquad = \mathbb{E}_{\mathcal{D}}[\sum_{i \in [d]} \pi(x)_i \cdot y_i] \\ &\qquad \geq \sum_{\tau \in \mathcal{T}} \sum_{i \in [d]} \Pr[\pi(x)_i \in \tau] \\ &\qquad\qquad \cdot \mathbb{E}_{\mathcal{D}}[\tau_1 \cdot y_i - |\tau_1 - \pi(x)_i| \cdot |y_i| \; | \pi(x)_i \in \tau] \\ &\qquad \geq \sum_{\tau \in \mathcal{T}} \sum_{i \in [d]} \Pr[\pi(x)_i \in \tau] \\ &\qquad\qquad \cdot \left(\mathbb{E}_{\mathcal{D}}[\tau_1 \cdot y_i | \pi(x)_i \in \tau] - \frac{wM}{2}\right) \\ &\qquad \geq \sum_{\tau \in \mathcal{T}} \sum_{i \in [d]} \Pr[\pi(x)_i \in \tau] \cdot \\ &\qquad\qquad \left(\mathbb{E}_{\mathcal{D}}[\tau_1 \cdot h(x)_i | \pi(x)_i \in \tau] \right. \\ &\qquad\qquad\qquad \left. - \frac{\alpha}{\Pr[\pi(x)_i \in \tau]} - \frac{wM}{2}\right)\end{aligned}$$

$$
\begin{aligned}
&\geq \sum_{\tau\in\mathcal{T}}\sum_{i\in[d]} \Pr[\pi(x)_i \in \tau]\cdot \\
&\quad \Bigg(\mathbb{E}_{\mathcal{D}}[\pi(x)_i \cdot h(x)_i \\
&\qquad - |\tau_1 - \pi(x)_i| \cdot |h(x)_i| \mid \pi(x)_i \in \tau] \\
&\qquad - \frac{\alpha}{\Pr[\pi(x)_i \in \tau]} - \frac{wM}{2}\Bigg) \\
&\geq \sum_{\tau\in\mathcal{T}}\sum_{i\in[d]} \Pr[\pi(x)_i \in \tau]\cdot \\
&\quad \Bigg(\mathbb{E}_{\mathcal{D}}[\pi(x)_i \cdot h(x)_i | \pi(x)_i \in \tau] \\
&\qquad - \frac{\alpha}{\Pr[\pi(x)_i \in \tau]} - wM\Bigg) \qquad\qquad = \mathbb{E}_{\mathcal{D}}[\pi(x)\cdot h(x)] - \alpha|\mathcal{T}|d - wMd, \\
&= \mathbb{E}_{\mathcal{D}}[\pi(x)\cdot h(x)] - \frac{\alpha d}{w} - wMd,
\end{aligned}
$$

where the first and fourth inequalities hold by the triangle inequality, the second and fifth by the assumption that $\|y\|_\infty \leq M$ for all $y \in \mathcal{Y}$, and the third inequality follows from h satisfying consistency condition on policy π. Setting $w = \sqrt{\alpha/M}$, we have $\mathbb{E}_{\mathcal{D}}[\pi(x)\cdot y] \geq \mathbb{E}_{\mathcal{D}}[\pi(x)\cdot h(x)] - 2d\sqrt{\alpha M}$. The reverse direction holds similarly. ◀

Proof of Lemma 11.

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\pi_h(x)\cdot y] \\
&\quad \geq \mathbb{E}_{\mathcal{D}}[\pi_h(x)\cdot h(x)] - 2d\sqrt{\alpha M} \\
&\quad \geq \mathbb{E}_{\mathcal{D}}[\pi(x)\cdot h(x)] - 2d\sqrt{\alpha M} \\
&\quad = \mathbb{E}_{\mathcal{D}}[\sum_{i\in[d]} \pi(x)_i \cdot h(x)_i] - 2d\sqrt{\alpha M} \\
&\quad \geq \sum_{i\in[d]}\sum_{\tau\in\mathcal{T}} \Pr[\pi(x)_i \in \tau]\Big(\mathbb{E}_{\mathcal{D}}[\tau_1 \cdot h(x)_i \\
&\qquad - |\tau_1 - \pi(x)_i||h(x)_i| \mid \pi(x)_i \in \tau]\Big) - 2d\sqrt{\alpha M} \\
&\quad \geq \sum_{i\in[d]}\sum_{\tau\in\mathcal{T}} \Pr[\pi(x)_i \in \tau]\Big(\mathbb{E}_{\mathcal{D}}[\tau_1 \cdot h(x)_i | \pi(x)_i \in \tau] \\
&\qquad\qquad - \frac{wM}{2}\Big) - 2d\sqrt{\alpha M} \\
&\quad \geq \sum_{i\in[d]}\sum_{\tau\in\mathcal{T}} \Pr[\pi(x)_i \in \tau]\Big(\mathbb{E}_{\mathcal{D}}[\tau_1 \cdot y_i | \pi(x)_i \in \tau] \\
&\qquad\qquad - \frac{\alpha}{\Pr[\pi(x)_i \in \tau]} - \frac{wM}{2}\Big) - 2d\sqrt{\alpha M}
\end{aligned}
$$

$$
\begin{aligned}
&\geq \sum_{i\in[d]}\sum_{\tau\in\mathcal{T}} \Pr[\pi(x)_i \in \tau]\Bigg(\mathbb{E}_{\mathcal{D}}[\pi(x)_i \cdot y_i \\
&\qquad - |\tau_1 - \pi(x)_i| \cdot |y_i| \mid \pi(x)_i \in \tau] \\
&\qquad - \frac{\alpha}{\Pr[\pi(x)_i \in \tau]} - \frac{wM}{2}\Bigg) - 2d\sqrt{\alpha M} \\
&\geq \sum_{i\in[d]}\sum_{\tau\in\mathcal{T}} \Pr[\pi(x)_i \in \tau]\Bigg(\mathbb{E}_{\mathcal{D}}[\pi(x)_i \cdot y_i | \pi(x)_i \in \tau] \\
&\qquad - \frac{\alpha}{\Pr[\pi(x)_i \in \tau]} - wM\Bigg) - 2d\sqrt{\alpha M}, \\
&= \sum_{i\in[d]}\sum_{\tau\in\mathcal{T}} \Pr[\pi(x)_i \in \tau]\mathbb{E}_{\mathcal{D}}[\pi(x)_i \cdot y_i | \pi(x)_i \in \tau] - \\
&\qquad \frac{\alpha d}{w} - wMd - 2d\sqrt{\alpha M},
\end{aligned}
$$

where the first inequality holds by Corollary 10, the second by the optimality of policy π_h with respect to h rather than the policy π, and the fifth by the consistency of h with respect to the policy π. Then, taking $w = \sqrt{\alpha/M}$, we have

$$
\begin{aligned}
&\sum_{i\in[d]}\sum_{\tau\in\mathcal{T}} \Pr[\pi(x)_i \in \tau]\mathbb{E}_{\mathcal{D}}[\pi(x)_i \cdot y_i | \pi(x)_i \in \tau] - \\
&\qquad \frac{\alpha d}{w} - wMd - 2d\sqrt{\alpha M} \\
&= \mathbb{E}_{\mathcal{D}}[\pi(x) \cdot r] - 4d\sqrt{\alpha M}.
\end{aligned}
$$

◀

Proof of Lemma 15. This proof follows similarly to that of Lemma 8. As stated in Lemmas 7 and 8, we know that between two adjacent rounds within the $\textsc{Update}(f, \cdot)$ procedure for some model f, the squared error of the model f drops by at least α^2. Therefore, we know that between two adjacent invocations of $\textsc{Update}(h^t, \cdot)$ and $\textsc{Update}(h^{t+1}, \cdot)$ within Algorithm 2, the squared error of model h^{t+1} must be at least α^2 less than the squared error of model h^t. Since $\mathcal{Y} \subseteq \mathbb{R}^d$ and $\|y\|_\infty \leq M$ for all $y \in \mathcal{Y}$, it must be the case that Algorithm 2 terminates after at most $\frac{dM^2}{\alpha^2}$ invocations of Algorithm 1. ◀

Proof of Lemma 16. This follows directly from k applications of Lemma 15, since Algorithm 2 maintains k separate predictors which are being iteratively updated using the $\textsc{Update}(\cdot, \cdot)$ procedure on an adaptively chosen sequence of conditioning events. ◀

Proof of Lemma 17.

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot y]\\
&\quad=\sum_{x\in\mathcal{X}}\Pr[X=x]\,(\pi_{\mathbf{h}}(x)\cdot\mathbb{E}[r|x])\\
&\quad=\sum_{x\in\mathcal{X}}\sum_{i\in[k]}\Pr[X=x]\mathbb{1}_{i=i^*(x)}\cdot\pi_{h_i}(x)\cdot\mathbb{E}[y|x]\\
&\quad\geq\sum_{i\in[k]}\sum_{\tau\in\mathcal{T}}\sum_{j\in[d]}\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\cdot\\
&\qquad\Big(\mathbb{E}_{\mathcal{D}}[\tau_1\cdot y_j-|\tau_1-\pi_{h_i}(x)_j|\cdot\\
&\qquad\quad|y_j|\;|\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\Big)\\
&\quad\geq\sum_{i\in[k]}\sum_{\tau\in\mathcal{T}}\sum_{j\in[d]}\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\cdot\\
&\qquad\Big(\mathbb{E}_{\mathcal{D}}[\tau_1\cdot y_j|\pi_{h_i}(x)_j\in\tau,i^*(x)=i]-\frac{wM}{2}\Big)\\
&\quad\geq\sum_{i\in[k]}\sum_{\tau\in\mathcal{T}}\sum_{j\in[d]}\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\cdot\\
&\qquad\Big(\mathbb{E}_{\mathcal{D}}[\tau_1\cdot h_i(x)_j|\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\\
&\qquad\quad-\frac{\alpha}{\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]}-\frac{wM}{2}\Big)\\
&\quad\geq\sum_{i\in[k]}\sum_{\tau\in\mathcal{T}}\sum_{j\in[d]}\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\cdot\\
&\qquad\Big(\mathbb{E}_{\mathcal{D}}[\pi_{h_i}(x)_j\cdot h_i(x)_j-|\tau_1-\pi_{h_i}(x)_j|\\
&\qquad\quad\cdot|h_i(x)_j|\;|\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\\
&\qquad\quad-\frac{\alpha}{\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]}-\frac{wM}{2}\Big)\\
&\quad\geq\sum_{i\in[k]}\sum_{\tau\in\mathcal{T}}\sum_{j\in[d]}\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\cdot\\
&\qquad\Big(\mathbb{E}_{\mathcal{D}}[\pi_{h_i}(x)_j\cdot h_i(x)_j|\pi_{h_i}(x)_j\in\tau,i^*(x)=i]\\
&\qquad\quad-\frac{\alpha}{\Pr[\pi_{h_i}(x)_j\in\tau,i^*(x)=i]}-wM\Big)\\
&\quad=\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot\mathbf{h}]-\alpha k|\mathcal{T}|d-wMd\\
&\quad=\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot\mathbf{h}]-\frac{\alpha kd}{w}-wMd,
\end{aligned}
$$

where the third inequality follows from the consistency conditions on the ensemble $\mathbf{h}$. Then, setting $w=\sqrt{\alpha k/M}$, we have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot\mathbf{h}]-\frac{\alpha kd}{w}-wMd\\
=\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot\mathbf{h}]-2d\sqrt{\alpha kM}.
\end{aligned}
$$

◀

Proof of Lemma 18.

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot y] &\geq \mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot \mathbf{h}(x)] - 2d\sqrt{\alpha k M}\\ &\geq \mathbb{E}_{\mathcal{D}}[\max_{j\in[k]} \pi_{h_j}(x)\cdot h_j(x)] - 2d\sqrt{\alpha k M},\end{aligned}$$

where the first inequality follows from Lemma 17 and the second inequality follows from Definition 13. ◀

Proof of Lemma 19.

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot y] &\geq \mathbb{E}_{\mathcal{D}}[\pi_{\mathbf{h}}(x)\cdot \mathbf{h}(x)] - 2d\sqrt{\alpha k M}\\ &\geq \mathbb{E}_{\mathcal{D}}[\pi_{h_{\phi(i^*(x))}}(x)\cdot h_{\phi(i^*(x))}(x)] - 2d\sqrt{\alpha k M}\\ &\geq \mathbb{E}_{\mathcal{D}}[\pi_{h_{\phi(i^*(x))}}(x)\cdot y] - 4d\sqrt{\alpha k M},\end{aligned}$$

where the first inequality follows from Lemma 17, the second inequality follows from Definition 13, and the third inequality follows from the "cross" consistency conditions, that model h_s is approximately consistent with respect to its own policy conditioned on model h_t having the highest self-evaluation, for all $s, t \in [k]$. ◀

Proof of Lemma 20. Fix $\pi \in \mathcal{P} \cup \{\pi_h\}, i \in [d], \tau \in \mathcal{T}$.

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}[\pi_h(x)\cdot r | \pi(x)_i \in \tau]\\ &= \mathbb{E}_{\mathcal{D}}[\sum_{j\in[d]} \pi_h(x)_j \cdot y_j | \pi(x)_i \in \tau]\\ &= \sum_{j\in[d]} \mathbb{E}_{\mathcal{D}}[\pi_h(x)_j \cdot y_j | \pi(x)_i \in \tau]\\ &\geq \sum_{j\in[d]} \mathbb{E}_{\mathcal{D}}[\tau_1 \cdot y_j - |\tau_1 - \pi_h(x)_j| \cdot |y_j| \ | \pi(x)_i \in \tau]\\ &\geq \sum_{j\in[d]} \left(\mathbb{E}_{\mathcal{D}}[\tau_1 \cdot y_j | \pi(x)_i \in \tau] - \frac{wM}{2}\right)\\ &\geq \sum_{j\in[d]} \mathbb{E}_{\mathcal{D}}[\tau_1 \cdot h(x) | \pi(x)_i \in \tau] - \frac{wMd}{2} - \frac{\alpha d}{\Pr[\pi(x)_i \in \tau]}\\ &\geq \mathbb{E}_{\mathcal{D}}[\pi_h(x)\cdot h(x) | \pi(x)_i \in \tau] - wMd - \frac{\alpha d}{\Pr[\pi(x)_i \in \tau]}\\ &\geq \mathbb{E}_{\mathcal{D}}[\pi(x)\cdot h(x) | \pi(x)_i \in \tau] - wMd - \frac{\alpha d}{\Pr[\pi(x)_i \in \tau]}\\ &\geq \mathbb{E}_{\mathcal{D}}[\tau_1 \cdot h(x) | \pi(x)_i \in \tau] - \frac{3wMd}{2} - \frac{\alpha d}{\Pr[\pi(x)_i \in \tau]}\\ &\geq \mathbb{E}_{\mathcal{D}}[\tau_1 \cdot y | \pi(x)_i \in \tau] - \frac{3wMd}{2} - \frac{2\alpha d}{\Pr[\pi(x)_i \in \tau]}\\ &\geq \mathbb{E}_{\mathcal{D}}[\pi(x)\cdot y | \pi(x)_i \in \tau] - 2wMd - \frac{2\alpha d}{\Pr[\pi(x)_i \in \tau]}\end{aligned}$$

where the third and penultimte inequalities follow from the consistency of h to policy π and the fifth from the pointwise optimality of policy π_h to model h. ◀