DEFENSIVE BOOSTING FOR ONLINE PROBABILISTIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

We study online probabilistic forecasting of binary outcomes chosen by an adaptive adversary. Given an online learning algorithm for a weak hypothesis class $\mathcal{H}$, we would like to efficiently obtain two incomparable guarantees that existing online boosting techniques provide separately. Online gradient boosting competes in Brier score with the best predictor induced by the span of $\mathcal{H}$ on every sequence — but promises nothing when the span does not contain an accurate predictor. Online weak-to-strong boosting drives classification error to zero under a weak-learning condition, but promises little when that condition fails. We give a simple defensive forecasting algorithm, the *Defensive Booster*, that obtains both guarantees. On every adaptive sequence, its Brier score is competitive with the best prediction induced by the span of $\mathcal{H}$ at the same rate as online gradient boosting; simultaneously, whenever the realized transcript satisfies the smooth weak-learning condition, its Brier score and randomized classification error satisfy the same rate guarantee as online classification boosting. This is achieved by operationalizing the *dual view* of boosting: when the Defensive Booster's randomized classification error is persistently high, its mistake weights form a smooth reweighting on which every weak hypothesis has low edge, yielding an ex-post *hard-core* certificate that the weak-learning condition fails. We also develop a strongly adaptive variant, which satisfies both guarantees and provides local hard-core certificates on every time interval. The Defensive Booster accesses just one weak-class learner, whereas the prior online boosting methods we compare against maintain large weak-learner ensembles. Experiments on synthetic and real data streams demonstrate strong predictive performance with 20–70$\times$ speedups over the individual boosting baselines.

1 INTRODUCTION

On each round t of an online probabilistic forecasting problem, an adversary reveals a context $x_t \in \mathcal{X}$, the learner announces a probability $p_t \in [0,1]$ for the binary outcome $Y_t \in \{0,1\}$, and the outcome is then revealed. The sequence may be arbitrary and adaptive to the learner's past predictions; we make no distributional assumptions. The forecast is scored by the Brier score $(Y_t - p_t)^2$: the squared-error proper scoring rule, minimized in expectation by the true conditional probability that $Y_t = 1$.

We study this problem through the lens of boosting. The learner is given an online learning algorithm for simple, "weak" probability predictors and wants to aggregate the weak class into stronger probability forecasts. To put two familiar boosting guarantees in common notation, encode a weak prediction $q(x) \in [0,1]$ as $h(x) = 2q(x) - 1 \in [-1,1]$, and let $\mathcal{H}$ denote the resulting class; likewise, encode Y_t as $\sigma_t = 2Y_t - 1$.

Online gradient boosting (Beygelzimer et al., 2015a) treats boosting as online convex optimization over combinations of weak hypotheses. Under squared loss, it guarantees Brier score competitive with the best predictor in the convex hull of $\mathcal{H}$, or more generally in a norm-bounded span. This guarantee holds on every sequence, but there need not be an accurate predictor in the span.

Online weak-to-strong boosting (Chen et al., 2012; Beygelzimer et al., 2015b) instead obtains the "AdaBoost phenomenon" under a smooth weak-learning condition: every sufficiently smooth reweighting of the examples admits a hypothesis with edge γ over random guessing. The edge of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

h under weights w_t is its normalized weighted correlation $|\sum_t w_t \sigma_t h(x_t)| / \sum_t w_t$; for a binary-valued weak learner, edge γ is equivalent, after possibly flipping its sign, to weighted classification error at most $(1-\gamma)/2$. When the condition holds down to the smoothness needed for the target accuracy, boosting drives classification error to zero. But when it fails, these algorithms promise little, and their natural output is a weighted vote rather than a probability.

This paper asks whether a single efficient online algorithm, outputting probability forecasts, can enjoy both guarantees at once: the unconditional comparator guarantee of gradient boosting and the conditional weak-to-strong guarantee of classification boosting.

1.1 Our results

We answer affirmatively with the *Defensive Booster* (Algorithm 1), built as a black-box reduction from an online learning algorithm for $\mathcal{H}$. Write $B_T = T^{-1} \sum_{t \le T} (Y_t - p_t)^2$ for its average Brier score.

First, we unconditionally obtain the same guarantee as online gradient boosting, at the same rate.

Informal Theorem (Span guarantee; Theorem 3.1)**.** *On every adaptive sequence, for every $f = \sum_j \alpha_j h_j$ with $h_j \in \mathcal{H}$ and $\sum_j |\alpha_j| \le \Lambda$, define $q_f(x) = (1 + f(x))/2$. Then*

$$B_T \le \frac{1}{T} \sum_{t=1}^{T} \left(Y_t - q_f(x_t)\right)^2 + O\left(\frac{1+\Lambda}{\sqrt{T}}\right).$$

The score q_f may be unrestricted outside $[0,1]$. The actual bound is second order: its regret term scales with the forecaster's own Brier score, yielding a fast $O(1/T)$ bound in the realizable span case (Corollary 3.2). Simultaneously, we obtain the guarantee of online weak-to-strong boosting, with the γ, ε dependence that is optimal in the weak-online-learning model of Beygelzimer et al. (2015b).

Informal Theorem (Weak-to-strong guarantee; Corollary 3.4)**.** *Fix $\rho, \gamma \in (0,1]$. If every $w \in [0,1]^T$ with $T^{-1} \sum_t w_t \ge \rho$ admits some $h \in \mathcal{H}$ such that $|\sum_t w_t \sigma_t h(x_t)| / \sum_t w_t \ge \gamma$, then the forecaster's Brier score and randomized classification error $T^{-1} \sum_t |Y_t - p_t|$ are at most $\max\{\rho, \widetilde{O}(1/(\gamma^2 T))\}$. Consequently, if the condition holds with $\rho \le \varepsilon$, both errors are at most ε after $T = \widetilde{O}(1/(\gamma^2 \varepsilon))$ rounds.*

Thresholding each forecast at $1/2$ gives a deterministic classifier whose average $0/1$ error is at most twice the randomized error.

The weak-to-strong guarantee has a complementary certificate (Theorem 3.3). If the forecaster's error remains large for long enough, its mistake weights $w_t = |Y_t - p_t|$ form a *hard-core witness*: a smooth reweighting of the realized rounds on which every weak hypothesis has low edge. Thus persistent error explicitly certifies that the weak-learning condition fails on the realized transcript.

The two guarantees are genuinely different: neither implies the other, and prior work that provides either guarantee in isolation does not provide both.

Informal Proposition (Separation; Appendix B)**.** *Neither guarantee implies the other. In one direction, for arbitrarily small constants $\gamma > 0$, there are binary-valued weak classes and transcripts on which every reweighting has edge at least γ — so the weak-to-strong guarantee forces vanishing Brier score and randomized error — yet every score induced by the span has squared loss bounded below by a constant. Conversely, there are transcripts with an arbitrarily small-loss comparator in the span but a smooth reweighting on which every weak hypothesis has zero edge, so the smooth weak-learning condition fails.*

Both guarantees can also be made strongly adaptive. A variant using $O(\log T)$ active copies of the same weak-class oracle satisfies both guarantees, up to polylogarithmic factors, simultaneously on every contiguous interval (Section 4). On each interval it competes with the best span comparator for that interval; if the smooth weak-learning condition holds on the interval, it obtains the strong-learning guarantee there. It also localizes the certificate above: whenever error remains large on an interval, the mistake weights restricted to that interval form a local hard-core witness. Because

the interval may be chosen after observing the transcript, this identifies where and when the smooth weak-learning condition fails, even if the condition holds on the full sequence.

We also evaluate the Defensive Booster empirically (Section 5). The stronger baseline family depends on the instance: gradient boosting can substantially outperform weak-to-strong boosting, and vice versa. On every instance, the Defensive Booster is competitive with, and often outperforms, the stronger baseline, while the ensemble baselines take 20–66$\times$ as much time per round: it maintains one online weak learner, whereas they each maintain 100.

A natural alternative is to run comparison boosters from both families in parallel and combine their probability forecasts with an online Brier-loss aggregator. We include this method as a baseline. It must run every constituent booster, while the Defensive Booster runs one weak-class learner. Moreover, a Brier guarantee controls randomized classification error only through $T^{-1}\sum_t |Y_t - p_t| \le \sqrt{B_T}$, so an $O(\varepsilon)$ Brier guarantee yields only $O(\sqrt{\varepsilon})$ randomized error, whereas the Defensive Booster directly guarantees $O(\varepsilon)$ under the smooth weak-learning condition. The aggregation scheme also provides no hard-core witness.

Finally, the algorithm also handles arbitrary bounded real-valued outcomes, and its squared-loss span guarantee holds unchanged. Appendix E describes the extension and evaluates it on three chronological regression streams.

Figure 1 previews performance on the real binary and bounded-regression streams; Section 5 and the experimental appendices give the complete results and protocol.

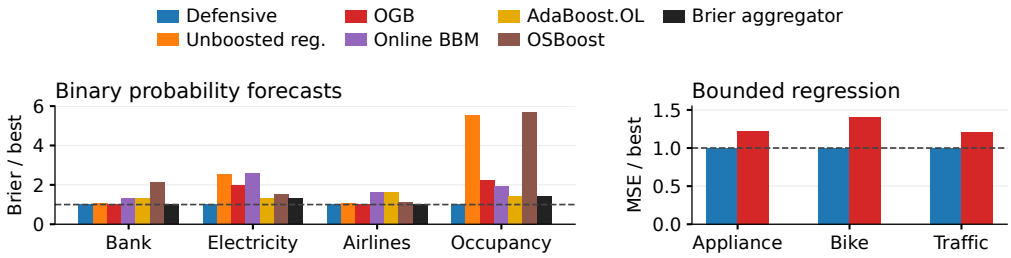

Figure 1: Real-stream preview. Each bar is loss divided by the best plotted method on that stream; lower is better. The Defensive Booster uses one weak learner, compared with 100 for each boosting baseline and 400 for the binary Brier aggregator.

1.2 Technique: Playing the Dual Side of the Boosting Game

The boosting game and its two views. In the batch setting, weak-to-strong boosting can be viewed as a zero-sum game between a *learner* player, who plays distributions over the weak class $\mathcal{H}$, and a *data* player, who plays reweightings of the dataset (Freund & Schapire, 1996). A winning strategy for the learner player is a distribution over weak hypotheses whose weighted majority vote attains perfect classification. A winning strategy for the data player is a *hard-core distribution*: a reweighting of the data on which no weak hypothesis has nontrivial edge (Impagliazzo, 1995).

Existing online weak-to-strong boosting operationalizes the *primal* view: run many copies of the weak learner in parallel and learn a weighted combination of their predictions (Chen et al., 2012; Beygelzimer et al., 2015b). We operationalize the *dual* view. Our forecaster never forms an ensemble: it maintains one online learner for $\mathcal{H}$ and two scalar adaptive-gradient states, for a per-round cost of one oracle call plus $O(1)$ arithmetic.

Defensive forecasting with orthogonality auditors. Write $\mu_t = 2p_t - 1$ for the forecast and $r_t = \sigma_t - \mu_t = 2(Y_t - p_t)$ for the residual. The design principle is *defensive forecasting* (Vovk et al., 2005b): the forecaster chooses p_t so that a designated family of statistical tests — *auditors* — cannot accumulate evidence that the forecasts differ from true probabilities. A weak-class auditor enforces multiaccuracy with respect to $\mathcal{H}$: hypotheses $h \in \mathcal{H}$ should have small empirical correlation with the residuals. A self-auditor enforces self-orthogonality: the forecast μ_t itself should have small empirical correlation with its own residuals. Self-orthogonality is implied by, but substantially

weaker than, calibration. This distinction matters because adversarial calibration is impossible online at the rates we seek (Qiao & Valiant, 2021; Dagan et al., 2025). On each round, the forecast is chosen by a one-dimensional root rule so that the aggregated auditor's realized gain is nonpositive; this is a deterministic one-dimensional instance of the framework of Farina & Perdomo (2026). The resulting certificate is

$$\sup_{h\in\mathcal{H}}\left|\sum_t h(x_t)r_t\right| \le A_H\sqrt{S_T}+B_H, \qquad \left|\sum_t \mu_t r_t\right| \le A_S\sqrt{S_T}+B_S, \quad S_T=\sum_t r_t^2.$$

The constants come from the online-learning primitives (Theorem 2.3).

High error yields a smooth hard-core witness. Identify p_t with the randomized classifier that predicts 1 with probability p_t. Its conditional mistake probability is $w_t = |Y_t - p_t|$, and binary outcomes give the key identity $w_t\sigma_t = r_t/2$. The average weight is exactly the randomized classification error, while the identity converts multiaccuracy into an edge bound for the mistake weighting. If the error is large enough for w to be smooth, this contradicts the smooth weak-learning condition. By contrapositive if the weak learning condition holds error is low. The self-bounding term $\sqrt{S_T}$ produces the target $1/(\gamma^2\varepsilon)$ rate. Self-orthogonality is not used in this part of the argument.

Self-orthogonality gives the span guarantee. Multiaccuracy with respect to $\mathcal{H}$ together with self-orthogonality gives squared error competitive with every model in the span of $\mathcal{H}$: these are the first-order optimality conditions for squared loss (see, e.g., their use in Kearns et al., 2025). For a span comparator f, the excess squared loss is controlled by the residuals against f, which multiaccuracy bounds, and the residuals against the forecasts themselves, which self-orthogonality bounds. This gives the span guarantee on every sequence (Theorem 3.1).

2 The Defensive Booster and its Certificate

We use the signed variables σ_t, μ_t, r_t defined above. The weak class $\mathcal{H}$ consists of functions $h : \mathcal{X} \to [-1, 1]$ and is symmetric: if $h \in \mathcal{H}$, then $-h \in \mathcal{H}$. Otherwise one replaces $\mathcal{H}$ by $\mathcal{H} \cup (-\mathcal{H})$.

Definition 2.1 (Multiaccuracy and self-orthogonality)**.** *The forecasts are* α-multiaccurate with respect to $\mathcal{H}$ *if*

$$\sup_{h\in\mathcal{H}}\left|\sum_{t=1}^{T} h(x_t)r_t\right| \le \alpha.$$

They are β-self-orthogonal *if*

$$\left|\sum_{t=1}^{T} \mu_t r_t\right| \le \beta.$$

Multiaccuracy tests the residuals against an external class of functions; self-orthogonality tests them against the forecast itself.

For $\Lambda \ge 0$, define

$$\mathrm{span}_\Lambda(\mathcal{H}) = \left\{\sum_{j=1}^{m}\alpha_j h_j : h_j \in \mathcal{H}, \sum_j |\alpha_j| \le \Lambda\right\}, \qquad q_f(x) = \frac{1+f(x)}{2}.$$

The score q_f need not lie in $[0, 1]$; the span theorem competes with unrestricted affine scores.

The algorithm has one problem-dependent online primitive: a weak-class oracle for $\mathcal{H}$. We ask for a *second-order* regret guarantee, meaning that regret scales with the square root of the cumulative squared coefficients rather than with $\sqrt{T}$. The coefficients supplied to the oracle will be the forecaster's residuals, so this gives a certificate whose error scales with the residual energy itself. This self-bounding structure produces the $1/(\gamma^2\varepsilon)$ weak-to-strong sample complexity in Corollary 3.4; a first-order $\sqrt{T}$ guarantee would give only $1/(\gamma^2\varepsilon^2)$.

Algorithm 1 Defensive Booster

1: Initialize the weak-class learner and scalar learners S, A.
2: **for** $t = 1, 2, \ldots$ **do**
3: Observe x_t; obtain $\widehat{h}_t$ from the weak-class learner, θ_t from S, and λ_t from A.
4: Set $q_{H,t} = (1 + \lambda_t)/2$, $q_{S,t} = (1 - \lambda_t)/2$, and $F_t(\mu) = q_{H,t}\widehat{h}_t + q_{S,t}\theta_t\mu$.
5: Set $\mu_t = \mathrm{Root}(F_t)$ and forecast $p_t = (1 + \mu_t)/2$.
6: Observe Y_t and set $r_t = 2Y_t - 1 - \mu_t$, $z_{H,t} = \widehat{h}_t r_t$, and $z_{S,t} = \theta_t\mu_t r_t$.
7: Update the weak learner with $c_t = r_t$, S with $g_t = \mu_t r_t$, and A with $g_t = (z_{H,t} - z_{S,t})/2$.
8: **end for**

After seeing x_t and before seeing $c_t \in [-2, 2]$, it outputs $\widehat{h}_t \in [-1, 1]$ and guarantees

$$\sup_{h\in\mathcal{H}} \sum_{t=1}^{T} c_t h(x_t) - \sum_{t=1}^{T} c_t \widehat{h}_t \le a_{\mathcal{H}} \sqrt{\sum_{t=1}^{T} c_t^2} + b_{\mathcal{H}}. \tag{1}$$

Finite experts, online linear optimization, and online RKHS regression give standard instantiations; Appendix A.1 gives precise examples and citations. We also use two copies of a scalar routine that, for coefficients $g_t \in [-2, 2]$, outputs $a_t \in [-1, 1]$ and satisfies

$$\sup_{a\in[-1,1]} \sum_{t=1}^{T} (a - a_t) g_t \le 4 \sqrt{\sum_{t=1}^{T} g_t^2} + 8. \tag{2}$$

Adaptive projected gradient descent provides this guarantee (Appendix A.2).

For continuous $F : [-1, 1] \to \mathbb{R}$, let $\mathrm{Root}(F)$ be any zero of F when one exists; if F is everywhere positive or negative, let it be 1 or -1, respectively.

Lemma 2.2 (Root sign property)**.** *For every continuous $F : [-1, 1] \to \mathbb{R}$, every $\sigma \in [-1, 1]$, and $\mu = \mathrm{Root}(F)$,*

$$F(\mu)(\sigma - \mu) \le 0. \tag{3}$$

Proof. The product is zero at a root. If F is positive throughout, then $\mu = 1$ and $\sigma - \mu \le 0$; if F is negative throughout, then $\mu = -1$ and $\sigma - \mu \ge 0$. □

The two scalar states in Algorithm 1 are denoted S for self-orthogonality and A for auditor aggregation.

The affine root takes constant time, so each round uses one weak-class prediction and update plus $O(1)$ arithmetic. The following certificate is the common source of both boosting guarantees.

The weak-class and self auditors have gains $z_{H,t} = \widehat{h}_t r_t$ and $z_{S,t} = \theta_t\mu_t r_t$. The scalar state A chooses their convex weights, while the root rule makes the resulting weighted gain nonpositive on every round, regardless of the label. Second-order regret therefore forces each auditor's cumulative gain to be small. The weak-class oracle transfers this to multiaccuracy, while the scalar state S transfers the self-auditor bound to self-orthogonality.

Theorem 2.3 (Second-order multiaccuracy and self-orthogonality)**.** *For every adaptive binary sequence, let $S_T = \sum_{t=1}^{T} r_t^2 = 4\sum_t (Y_t - p_t)^2$. Algorithm 1 satisfies*

$$\sup_{h\in\mathcal{H}} \left| \sum_{t=1}^{T} h(x_t) r_t \right| \le A_H \sqrt{S_T} + B_H, \qquad \left| \sum_{t=1}^{T} \mu_t r_t \right| \le A_S \sqrt{S_T} + B_S,$$

where $A_H = a_{\mathcal{H}} + 4$, $B_H = b_{\mathcal{H}} + 8$, $A_S = 8$, and $B_S = 16$.

Proof. Let $\bar{z}_t = (z_{H,t} + z_{S,t})/2$ and $v_t = (z_{H,t} - z_{S,t})/2$. The gain selected by A is

$$\bar{z}_t + \lambda_t v_t = q_{H,t} z_{H,t} + q_{S,t} z_{S,t} = F_t(\mu_t) r_t \le 0$$

by (3). Applying (2) to A with comparators 1 and -1, and using $|v_t| \le |r_t|$, gives

$$\sum_t z_{H,t}, \quad \sum_t z_{S,t} \le 4\sqrt{S_T} + 8. \tag{4}$$

Equation (1), used with $c_t = r_t$, now bounds $\sum_t h(x_t) r_t$ by $(a_{\mathcal{H}} + 4)\sqrt{S_T} + b_{\mathcal{H}} + 8$; symmetry gives the absolute value. Finally, applying (2) to S, using (4) and $|\mu_t r_t| \le |r_t|$, gives $\sup_{\theta \in [-1,1]} \sum_t \theta \mu_t r_t \le 8\sqrt{S_T} + 16$, which is the self-orthogonality bound. □

3 Two consequences of the certificate

This section derives the paper's guarantees from the two inequalities in Theorem 2.3. They are parallel consequences, not consequences of one another: the hard-core statement uses multiaccuracy alone, whereas the span statement also uses self-orthogonality.

3.1 Span regret

Theorem 3.1 (Brier/span guarantee). *For every $f \in \mathrm{span}_\Lambda(\mathcal{H})$, Algorithm 1 satisfies*

$$B_T \le \frac{1}{T}\sum_{t=1}^{T}(Y_t - q_f(x_t))^2 + \frac{\Lambda A_H + A_S}{\sqrt{T}}\sqrt{B_T} + \frac{\Lambda B_H + B_S}{2T}.$$

Consequently, its regret to every $f \in \mathrm{span}_\Lambda(\mathcal{H})$ is at most $(\Lambda A_H + A_S)/\sqrt{T} + (\Lambda B_H + B_S)/(2T)$.

Proof. Convexity of squared loss gives $(\sigma_t - \mu_t)^2 \le (\sigma_t - f(x_t))^2 + 2r_t(f(x_t) - \mu_t)$. For $f = \sum_j \alpha_j h_j$ with $\sum_j |\alpha_j| \le \Lambda$, Theorem 2.3 bounds the two correlations in the sum by $\Lambda(A_H\sqrt{S_T} + B_H)$ and $A_S\sqrt{S_T} + B_S$. Summing and using $S_T = 4TB_T$ and $(\sigma_t - f(x_t))^2 = 4(Y_t - q_f(x_t))^2$ proves the first claim; $B_T \le 1$ proves the second. □

Corollary 3.2 (Low-loss span guarantee). *If $B_f = T^{-1}\sum_t (Y_t - q_f(x_t))^2$, then*

$$B_T \le B_f + C\sqrt{B_f/T} + \frac{3C^2}{2T} + \frac{3D}{4T}, \quad C = \Lambda A_H + A_S, \quad D = \Lambda B_H + B_S. \tag{5}$$

In particular, $B_f = 0$ implies $B_T = O((C^2 + D)/T)$.

The algebra appears in Appendix A.3.

3.2 Hard-core witness and weak-to-strong boosting

A reweighting is a sequence $w \in [0,1]^T$ with density $\rho(w) = T^{-1}\sum_t w_t$. When $\rho(w) > 0$, define

$$\mathrm{edge}_{\mathcal{H}}(w) = \sup_{h \in \mathcal{H}} \frac{|\sum_t w_t \sigma_t h(x_t)|}{\sum_t w_t}.$$

For $\rho_0, \gamma_0 \in (0,1]$, the transcript satisfies the (ρ_0, γ_0)-*smooth weak-learning condition* if $\mathrm{edge}_{\mathcal{H}}(w) \ge \gamma_0$ for every w with $\rho(w) \ge \rho_0$. For binary-valued hypotheses, this says that, after a possible sign flip, some weak classifier has weighted error at most $(1 - \gamma_0)/2$. Appendix A.4 relates this definition to the smooth distributions used in classical boosting.

Theorem 3.3 (Hard-core mistake weighting). *Let $w_t = |Y_t - p_t|$ and $\rho_w = T^{-1}\sum_t w_t$. Then*

$$B_T = \frac{1}{T}\sum_t w_t^2 \le \rho_w, \qquad \left|\frac{1}{T}\sum_t w_t \sigma_t h(x_t)\right| \le \frac{A_H\sqrt{TB_T} + B_H/2}{T} \quad \textit{for every } h \in \mathcal{H}.$$

Consequently, if $\rho_w > 0$ then

$$\mathrm{edge}_{\mathcal{H}}(w) \le \frac{A_H\sqrt{TB_T} + B_H/2}{T\rho_w}.$$

Proof. Binary outcomes give $(Y_t - p_t)^2 = w_t^2 \le w_t$ and the key identity

$$w_t \sigma_t = Y_t - p_t = \frac{\sigma_t - \mu_t}{2}.$$

The multiaccuracy inequality in Theorem 2.3, together with $S_T = 4TB_T$, gives the unnormalized correlation bound. Dividing by ρ_w gives the edge bound. □

Corollary 3.4 (Second-order weak-to-strong rate)**.** *If the transcript satisfies the (ρ_0, γ_0)-smooth weak-learning condition, then*

$$B_T, \rho_w \le \max\left\{\rho_0, \frac{4A_H^2}{\gamma_0^2 T}, \frac{B_H}{\gamma_0 T}\right\}.$$

The threshold classifier $\mathbf{1}\{p_t \ge 1/2\}$ has average error at most $2\rho_w$.

Proof. If $\rho_w < \rho_0$ the claim follows from $B_T \le \rho_w$. Otherwise the weak-learning condition gives $\mathrm{edge}_{\mathcal{H}}(w) \ge \gamma_0$, while Theorem 3.3 upper-bounds the same quantity. If ρ_w exceeded both remaining terms in the maximum, then, using $B_T \le \rho_w$, each term in the numerator would be strictly smaller than $\gamma_0 T \rho_w / 2$, a contradiction. Finally, every threshold mistake satisfies $|Y_t - p_t| \ge 1/2$. □

4 Boosting on every interval

Applying a standard geometric-cover wrapper to the weak-class and scalar learners yields a strongly adaptive Defensive Booster. It maintains at most $1 + \lceil \log_2 T \rceil$ active weak-class learners and satisfies both certificate inequalities simultaneously on every interval, with logarithmic overhead. Thus it competes with the best span comparator on each interval; if the (ρ, γ)-smooth weak-learning condition holds on an interval of length n, its Brier score and randomized error there are at most $\max\{\rho, \widetilde{O}(1/(\gamma^2 n))\}$. If error instead remains large, the mistake weights on that interval form a local smooth, low-edge witness. The interval may be selected after seeing the transcript. Appendix A.5 states the exact bounds and gives the reduction and proofs.

5 Experiments

We compare the Defensive Booster with online gradient boosting, online weak-to-strong boosting, and the more naive Brier aggregator strategy on controlled synthetic streams and four naturally ordered real datasets. The synthetic streams separately test settings in which the span contains an informative predictor and settings in which the smooth weak-learning condition holds. Across the binary experiments, the Defensive Booster's Brier loss is competitive with the best baseline on each dataset and often improves upon it substantially, while the ensemble baselines take 20–66× as much time per round.

Protocol. Two unboosted controls isolate the benefit of aggregation: one runs an online squared-loss regressor and the other an online classifier. Four ensemble baselines represent the two boosting traditions: OGB (Beygelzimer et al., 2015a), Online BBM and AdaBoost.OL (Beygelzimer et al., 2015b), and OSBoost (Chen et al., 2012). The Brier aggregator combines the forecasts of these four ensembles by exponential weighting under Brier loss. The Defensive Booster and each unboosted control maintain one learner over the weak class; each boosting baseline maintains $N = 100$, and the aggregator runs all four ensembles, for a total of 400.

We report Brier loss, randomized classification error $T^{-1}\sum_t |Y_t - p_t|$, and deterministic $0/1$ error after thresholding at $1/2$. We fix all hyperparameters before examining performance and do not tune them separately for each stream. Synthetic results use $T = 3000$ and average 20 seeds; each real dataset is processed once in recorded order, without shuffling. Appendix D.1 gives complete algorithm, tuning, preprocessing, and reproduction details.

Two synthetic streams favor complementary baseline families. In the *binary aggregation* stream, 100 hidden orientations of paired binary rules have margin at least .16 on every round, so every reweighting has a weak rule with edge .16 although no fixed affine span score has small Brier loss.

In the *random-label mixture*, a linear rule governs 65% of the rounds and the remaining labels are independent bits; the linear span is informative, but the random component supports a smooth low-edge weighting. Figure 2 asks whether one algorithm can track the appropriate boosting family on both streams.

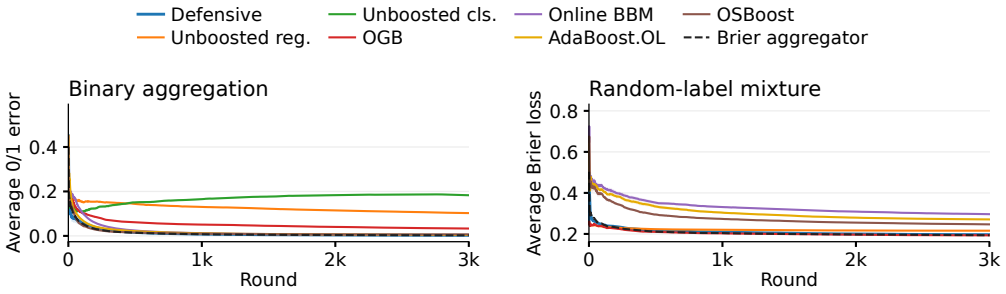

Figure 2: The Defensive Booster tracks the stronger baseline family on two complementary streams. Left: classification boosters approach zero error on the binary aggregation stream while OGB does not. Right: OGB approaches the informative span on the random-label mixture while classification boosters incur larger Brier loss. Curves average 20 seeds.

On binary aggregation, the Defensive Booster has classification error .0026, compared with .0041 for Online BBM, .0042 for AdaBoost.OL, .0068 for OSBoost, and .0331 for OGB. On the random-label mixture, its Brier loss is .1965, compared with .1933 for OGB and at least .2467 for each classification booster. The Brier aggregator reaches .1937 by running all four ensembles. Appendix D.2 reports standard errors, additional streams, further certificate diagnostics, and the learner-count sweep.

The random-label mixture also makes the hard-core certificate observable. Figure 3 shows that the Defensive Booster's normalized multiaccuracy and self-orthogonality errors decay. Its mistake weighting retains nontrivial density while the weak class's edge under that weighting decreases, as Theorem 3.3 predicts.

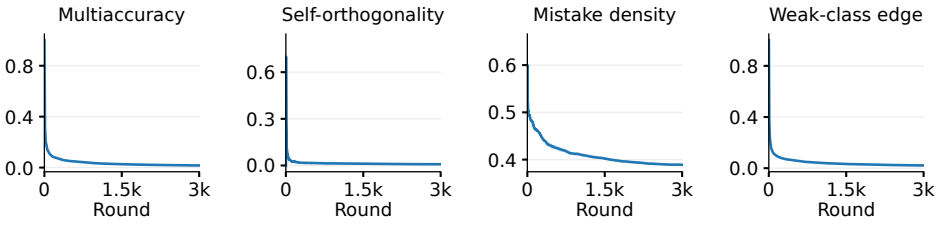

Figure 3: Certificate diagnostics on the random-label mixture, averaged over 20 seeds. The first two panels show cumulative multiaccuracy and self-orthogonality correlations divided by the current round; the final two show the density and weak-class edge of the mistake weighting.

We also evaluate four public binary streams in their recorded order: Bank Marketing predicts term-deposit subscription (Moro et al., 2014); Electricity predicts price movement (Harries, 1999; MOA, 2011); Airlines predicts flight delays (MOA, 2011); and Occupancy predicts office occupancy from sensors (Candanedo & Feldheim, 2016). All methods use the same 128-dimensional hashed features and unit-ball linear weak class. Table 1 shows that the Defensive Booster substantially improves upon every baseline on Electricity and Occupancy, essentially ties the best methods on Airlines, and is within .0010 of the aggregator on Bank.

Dataset	Defensive	Unboosted reg.	Unboosted cls.	OGB	BBM	AdaBoost.OL	OSBoost	Aggregator
Bank	.0800	.0845	.1091	.0791	.1026	.1046	.1679	**.0790**
Electricity	**.0772**	.1957	.3669	.1516	.2010	.1007	.1167	.1006
Airlines	.2094	.2190	.3901	**.2094**	.3411	.3417	.2354	.2094
Occupancy	**.0071**	.0396	.0961	.0159	.0138	.0103	.0406	.0101

Table 1: Average online Brier loss on naturally ordered binary streams. Hard-label methods are evaluated as $\{0, 1\}$ probability forecasts. Bold marks the lowest unrounded value.

Across all binary streams, the 100-learner ensembles take 20–66 times as much wall-clock time per round as the Defensive Booster. The appendix also reports that the bounded-outcome version improves normalized squared loss over 100-stage OGB on three chronological regression datasets, and that the strongly adaptive variant improves performance under several forms of distribution shift.

6 Related work

Online gradient boosting. Beygelzimer et al. (2015a) start from an online linear-loss learner for $\mathcal{H}$ and use N copies to compete with $\mathrm{conv}(\mathcal{H})$ or a norm-bounded span under smooth convex losses. Our Brier/span guarantee is closest to this result in the squared-loss case: it competes with unrestricted real-valued scores in a norm-bounded span. Under the same online-linear-oracle primitive, however, the Defensive Booster uses one weak-class learner rather than an N-stage ensemble.

Online weak-to-strong boosting. The closest classification predecessor is Chen et al. (2012), who adapt SmoothBoost to online binary boosting using smooth distributions. Beygelzimer et al. (2015b) give the rate-optimal online boosting algorithm, Online BBM, under weak online learnability assumptions. In their model, the optimal sample-complexity dependence for error ε is $1/(\gamma^2\varepsilon)$ up to logarithmic and excess-loss terms; our algorithm matches this dependence.

These papers make different weak-oracle assumptions. Chen et al. and Beygelzimer et al. assume that the online learner's own predictions have a fixed positive edge over random guessing on every admissible stream, up to an excess-loss term. Brukhim et al. (2020) instead assume a multiplicative agnostic oracle and boost it to regret against the best $h \in \mathcal{H}$. Our primitive is a no-regret learner for $\mathcal{H}$ under linear losses: it need not have any absolute edge, but it competes with every $h \in \mathcal{H}$ on the realized residual losses. Here $\mathcal{H}$ supplies weak directions that are aggregated into forecasts competing with $\mathrm{span}(\mathcal{H})$; positive edge enters separately through the ex-post smooth weak-learning condition.

Multiaccuracy and hard-core measures. Multiaccuracy was introduced as a black-box correction criterion by Kim et al. (2019), following multicalibration (Hébert-Johnson et al., 2018). Its connection to hard-core measures appears in complexity-theoretic regularity lemmas (Trevisan et al., 2009); stronger conditions yield related hard-core conclusions (Casacuberta et al., 2024; 2025). Our hard-core theorem needs multiaccuracy alone, while self-orthogonality is used only for span regret. This distinction matters online because full adversarial calibration cannot generally achieve the $O(\sqrt{T})$ scale of our certificate (Qiao & Valiant, 2021; Dagan et al., 2025).

Defensive forecasting. Defensive forecasting chooses probabilities that prevent continuous tests from accumulating positive gain (Vovk et al., 2005b;a). Our affine root step is a deterministic one-dimensional instance of the online-learning and variational-inequality framework of Farina & Perdomo (2026). Its additional structure turns one certificate into both span regret and a hard-core mistake weighting. Appendix C discusses the broader literatures on boosting as loss optimization, strong adaptivity, smooth boosting and hard-core sets, multicalibration, and defensive forecasting.

AI use statement

The authors used OpenAI GPT 5.6 Pro and GPT 5.6 in the Codex environment for development and checking of mathematical arguments, drafting and editing, software implementation, and analysis. The authors reviewed and verified the final claims, proofs, citations, code, and results and take responsibility for the manuscript.

REFERENCES

Boaz Barak, Moritz Hardt, and Satyen Kale. The uniform hardcore lemma via approximate bregman projections. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1193–1200. SIAM, 2009. doi: 10.1137/1.9781611973068.129. URL https://epubs.siam.org/doi/10.1137/1.9781611973068.129.

Osbert Bastani, Varun Gupta, Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Practical adversarial multivalid conformal prediction. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 29362–29373. Curran Associates, Inc., 2022. doi: 10.52202/068431-2129. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/bcdaaa1aec3ae2aa39542acefdec4e4b-Abstract-Conference.html.

Alina Beygelzimer, Elad Hazan, Satyen Kale, and Haipeng Luo. Online gradient boosting. In *Advances in Neural Information Processing Systems 28*, pp. 2458–2466, 2015a. URL https://proceedings.neurips.cc/paper/2015/hash/0a1bf96b7165e962e90cb14648c9462d-Abstract.html.

Alina Beygelzimer, Satyen Kale, and Haipeng Luo. Optimal and adaptive algorithms for online boosting. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2323–2331, Lille, France, 07–09 Jul 2015b. PMLR. URL https://proceedings.mlr.press/v37/beygelzimer15.html.

Nataly Brukhim, Xinyi Chen, Elad Hazan, and Shay Moran. Online agnostic boosting via regret minimization. In *Advances in Neural Information Processing Systems 33*, pp. 644–654, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/07168af6cb0ef9f78dae15739dd73255-Abstract.html.

Luis Candanedo. Appliances energy prediction. UCI Machine Learning Repository, 2017. URL https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction.

Luis M. Candanedo and Véronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings*, 112:28–39, 2016. doi: 10.1016/j.enbuild.2015.11.071. URL https://doi.org/10.1016/j.enbuild.2015.11.071.

Sílvia Casacuberta, Cynthia Dwork, and Salil Vadhan. Complexity-theoretic implications of multicalibration. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pp. 1071–1082. ACM, 2024. doi: 10.1145/3618260.3649748. URL https://doi.org/10.1145/3618260.3649748.

Sílvia Casacuberta, Parikshit Gopalan, Varun Kanade, and Omer Reingold. How global calibration strengthens multiaccuracy. In *Proceedings of the 66th IEEE Symposium on Foundations of Computer Science*, pp. 1198–1227. IEEE, 2025. doi: 10.1109/FOCS63196.2025.00063. URL https://doi.org/10.1109/FOCS63196.2025.00063.

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2–3):321–352, 2007. doi: 10.1007/s10994-006-5001-7. URL https://link.springer.com/article/10.1007/s10994-006-5001-7.

Shang-Tse Chen, Hsuan-Tien Lin, and Chi-Jen Lu. An online boosting algorithm with theoretical justifications. In John Langford and Joelle Pineau (eds.), *Proceedings of the 29th International Conference on Machine Learning*, pp. 1007–1014. Omnipress, 2012. URL https://icml.cc/2012/papers/538.pdf.

Natalie Collina, Jiuyao Lu, Georgy Noarov, and Aaron Roth. Optimal lower bounds for online multicalibration. *arXiv preprint arXiv:2601.05245*, 2026. URL https://arxiv.org/abs/2601.05245.

Ashok Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2250–2259. PMLR, 2020. URL https://proceedings.mlr.press/v119/cutkosky20a.html.

Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, Noah Golowich, Robert Kleinberg, and Princewill Okoroafor. Breaking the $T^{2/3}$ barrier for sequential calibration. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pp. 2007–2018, 2025. doi: 10.1145/3717823.3718178. URL https://doi.org/10.1145/3717823.3718178.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1405–1411. PMLR, 2015. URL https://proceedings.mlr.press/v37/daniely15.html.

Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1095–1108. ACM, 2021. doi: 10.1145/3406325.3451064. URL https://doi.org/10.1145/3406325.3451064.

Hadi Fanaee-T. Bike sharing. UCI Machine Learning Repository, 2013. URL https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset.

Gabriele Farina and Juan Carlos Perdomo. An efficient black-box reduction from online learning to multicalibration, and a new route to Φ-regret minimization. *arXiv preprint arXiv:2604.19592*, 2026. URL https://arxiv.org/abs/2604.19592.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pp. 325–332. ACM Press, 1996. doi: 10.1145/238061.238163. URL https://doi.org/10.1145/238061.238163.

Pierre Gaillard, Gilles Stoltz, and Tim van Erven. A second-order bound with excess losses. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári (eds.), *Proceedings of the 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pp. 176–196. PMLR, 2014. URL https://proceedings.mlr.press/v35/gaillard14.html.

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*, pp. 2725–2792. SIAM, 2024. doi: 10.1137/1.9781611977912.98. URL https://epubs.siam.org/doi/10.1137/1.9781611977912.98.

Dmitry Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *Journal of Machine Learning Research*, 4:101–117, 2003. URL https://www.jmlr.org/papers/v4/gavinsky03a.html.

Rohan Ghuge, Vidya Muthukumar, and Sahil Singla. Improved and oracle-efficient online ℓ_1-multicalibration. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19437–19457. PMLR, 2025. URL https://proceedings.mlr.press/v267/ghuge25a.html.

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*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11459–11492. PMLR, 2023. URL https://proceedings.mlr.press/v202/globus-harris23a.html.

Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference*, volume 215 of *Leibniz International Proceedings in Informatics*, pp. 79:1–79:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.ITCS.2022.79. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.79.

Parikshit Gopalan, Lunjia Hu, Michael P. Kim, Omer Reingold, and Udi Wieder. Loss minimization through the lens of outcome indistinguishability. In *14th Innovations in Theoretical Computer Science Conference*, volume 251 of *Leibniz International Proceedings in Informatics*, pp. 60:1–60:20. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPIcs.ITCS.2023.60. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2023.60.

Varun Gupta, Christopher Jung, Georgy Noarov, Mallesh M. Pai, and Aaron Roth. Online multivalid learning: Means, moments, and prediction intervals. In *13th Innovations in Theoretical Computer Science Conference*, volume 215 of *Leibniz International Proceedings in Informatics*, pp. 82:1–82:24. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.ITCS.2022.82. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2022.82.

Michael Harries. Splice-2 comparative evaluation: Electricity pricing. Technical Report 9905, School of Computer Science and Engineering, University of New South Wales, 1999. URL https://cgi.cse.unsw.edu.au/~reports/.

Elad Hazan and Karan Singh. Boosting for online convex optimization. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4140–4149. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/hazan21a.html.

Ursula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1939–1948. PMLR, 2018. URL https://proceedings.mlr.press/v80/hebert-johnson18a.html.

John Hogue. Metro interstate traffic volume. UCI Machine Learning Repository, 2019. URL https://archive.ics.uci.edu/dataset/492/metro+interstate+traffic+volume.

Hanzhang Hu, Wen Sun, Arun Venkatraman, Martial Hebert, and Andrew Bagnell. Gradient boosting on stochastic data streams. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 595–603. PMLR, 2017. URL https://proceedings.mlr.press/v54/hu17a.html.

Lunjia Hu, Haipeng Luo, Spandan Senapati, and Vatsal Sharan. Efficient swap multicalibration of elicitable properties. In Steve Hanneke and Tor Lattimore (eds.), *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 3314–3348. PMLR, 29 Jun–03 Jul 2026. URL https://proceedings.mlr.press/v336/hu26b.html.

Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pp. 538–545. IEEE Computer Society, 1995. doi: 10.1109/SFCS.1995.492584. URL https://doi.org/10.1109/SFCS.1995.492584.

Michael Kearns, Aaron Roth, and Emily Ryu. Networked information aggregation via machine learning. *arXiv preprint arXiv:2507.09683*, 2025. URL https://arxiv.org/abs/2507.09683.

Michael P. Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254. ACM, 2019. doi: 10.1145/3306618.3314287. URL https://doi.org/10.1145/3306618.3314287.

Jyrki Kivinen, Alexander J. Smola, and Robert C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004. doi: 10.1109/TSP.2004.830991.

Adam R. Klivans and Rocco A. Servedio. Boosting and hard-core set construction. *Machine Learning*, 51(3):217–238, 2003. doi: 10.1023/A:1022949332276. URL https://link.springer.com/article/10.1023/A:1022949332276.

Llew Mason, Jonathan Baxter, Peter L. Bartlett, and Marcus R. Frean. Boosting algorithms as gradient descent. In *Advances in Neural Information Processing Systems 12*, pp. 512–518. MIT Press, 1999. URL https://proceedings.neurips.cc/paper/1999/hash/96a93ba89a5b5c6c226e49b88973f46e-Abstract.html.

MOA. Massive online analysis datasets. Dataset repository, August 2011. URL https://moa.cms.waikato.ac.nz/datasets/. Accessed 2026-07-02.

Sérgio Moro, Paulo Rita, and Paulo Cortez. Bank marketing. UCI Machine Learning Repository, 2014. URL https://archive.ics.uci.edu/dataset/222/bank+marketing.

Indraneel Mukherjee, Cynthia Rudin, and Robert E. Schapire. The rate of convergence of AdaBoost. *Journal of Machine Learning Research*, 14(70):2315–2347, 2013. URL https://www.jmlr.org/papers/v14/mukherjee13b.html.

Alexandru Niculescu-Mizil and Richard A. Caruana. Obtaining calibrated probabilities from boosting. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 413–420, 2005. URL https://arxiv.org/abs/1207.1403.

Georgy Noarov and Aaron Roth. The statistical scope of multicalibration. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26283–26310. PMLR, 2023. URL https://proceedings.mlr.press/v202/noarov23a.html.

Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46762–46783. PMLR, 2025. URL https://proceedings.mlr.press/v267/noarov25b.html.

Francesco Orabona and Dávid Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018. doi: 10.1016/j.tcs.2017.11.021. URL https://doi.org/10.1016/j.tcs.2017.11.021.

Nikunj C. Oza and Stuart J. Russell. Online bagging and boosting. In Thomas S. Richardson and Tommi S. Jaakkola (eds.), *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, volume R3 of *Proceedings of Machine Learning Research*, pp. 229–236. PMLR, 2001. URL https://proceedings.mlr.press/r3/oza01a.html. Reissued by PMLR on 31 March 2021.

Juan Carlos Perdomo and Benjamin Recht. In defense of defensive forecasting. *arXiv preprint arXiv:2506.11848*, 2025. URL https://arxiv.org/abs/2506.11848.

Mingda Qiao and Gregory Valiant. Stronger calibration lower bounds via sidestepping. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 456–466, 2021. doi: 10.1145/3406325.3451050. URL https://doi.org/10.1145/3406325.3451050.

Rocco A. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003. URL https://www.jmlr.org/papers/v4/servedio03a.html.

Vinicius M. A. Souza, Denis M. dos Reis, André G. Maletzke, and Gustavo E. A. P. A. Batista. Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*, 34(6):1805–1858, 2020. doi: 10.1007/s10618-020-00698-5. URL https://doi.org/10.1007/s10618-020-00698-5.

Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Regularity, boosting, and efficiently simulating every high-entropy distribution. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, pp. 126–136. IEEE, 2009. URL https://people.seas.harvard.edu/~salil/research/regularity-abs.html.

Vladimir Vovk. Defensive forecasting for optimal prediction with expert advice. Technical Report Working Paper 20, Game-Theoretic Probability and Finance Project, August 2007. URL https://arxiv.org/abs/0708.1503.

Vladimir Vovk, Ilia Nouretdinov, Akimichi Takemura, and Glenn Shafer. Defensive forecasting for linear protocols. In *Algorithmic Learning Theory, 16th International Conference, ALT 2005*, volume 3734 of *Lecture Notes in Computer Science*, pp. 459–473. Springer, 2005a. doi: 10.1007/11564089_35.

Vladimir Vovk, Akimichi Takemura, and Glenn Shafer. Defensive forecasting. In Robert G. Cowell and Zoubin Ghahramani (eds.), *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, volume R5 of *Proceedings of Machine Learning Research*, pp. 365–372. PMLR, 2005b. URL https://proceedings.mlr.press/r5/vovk05a.html. Reissued by PMLR on 30 March 2021.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. Technical Report CMU-CS-03-110, Carnegie Mellon University, 2003. URL https://www.cs.cmu.edu/~maz/publications/techconvex.pdf.

A ALGORITHMIC DETAILS AND DEFERRED PROOFS

This appendix gives standard oracle instantiations, an explicit scalar second-order learner, and deferred derivations used in the main text.

A.1 ORACLE AND SCALAR LEARNER INSTANTIATIONS

For a finite symmetric class, a second-order experts algorithm gives $a_{\mathcal{H}} = O(\sqrt{\log|\mathcal{H}|})$ and $b_{\mathcal{H}} = O(\log|\mathcal{H}|)$ in (1) (Cesa-Bianchi et al., 2007). More generally, adaptive or scale-free online linear optimization over $\mathrm{conv}(\mathcal{H})$ gives data-dependent regret in terms of cumulative gradient norms (Zinkevich, 2003; Kivinen et al., 2004; Orabona & Pál, 2018). For example, for the RKHS ball $\mathcal{H}_B = \{x \mapsto \langle u, \phi(x)\rangle : \|u\| \le B\}$ with $\|\phi(x)\| \le \kappa$ and $B\kappa \le 1$, these methods give $O(B\kappa\sqrt{\sum_t c_t^2})$ regret up to lower-order terms.

The scalar learners in Algorithm 1 can be implemented by the following adaptive projected-gradient routine.

Definition A.1 (Scalar adaptive OGD)**.** *Initialize $a_1 = 0$ and $V_0 = 4$. On round t, output $a_t \in [-1,1]$ and, after observing $g_t \in [-2,2]$, update*

$$a_{t+1} = \Pi_{[-1,1]}\left(a_t + \frac{g_t}{\sqrt{V_{t-1}}}\right), \qquad V_t = V_{t-1} + g_t^2.$$

Lemma A.2 (Second-order scalar regret)**.** *For every $a \in [-1,1]$, scalar adaptive OGD satisfies*

$$\sum_{t=1}^{T}(a - a_t)g_t \le 4\sqrt{4 + \sum_{t=1}^{T} g_t^2} \le 4\sqrt{\sum_{t=1}^{T} g_t^2} + 8.$$

A.2 SCALAR SECOND-ORDER REGRET

Proof of Lemma A.2. Let $\eta_t = 1/\sqrt{V_{t-1}}$. Nonexpansiveness of projection gives

$$g_t(a - a_t) \le \frac{(a-a_t)^2 - (a-a_{t+1})^2}{2\eta_t} + \frac{\eta_t g_t^2}{2}.$$

Because η_t is nonincreasing and the diameter of $[-1,1]$ is 2, the sum of the first terms is at most $2/\eta_T \le 2\sqrt{V_T}$. Moreover, $g_t^2 \le 4 \le V_{t-1}$, and hence

$$\frac{g_t^2}{\sqrt{V_{t-1}}} = \left(1 + \sqrt{\frac{V_t}{V_{t-1}}}\right)\left(\sqrt{V_t} - \sqrt{V_{t-1}}\right) \le \frac{5}{2}\left(\sqrt{V_t} - \sqrt{V_{t-1}}\right).$$

Thus the sum of the second terms is at most $\frac{5}{4}\sqrt{V_T}$. Adding the two contributions gives

$$\sum_{t=1}^{T}(a - a_t)g_t \le \left(2 + \frac{5}{4}\right)\sqrt{V_T} = \frac{13}{4}\sqrt{V_T} \le 4\sqrt{V_T}.$$

Finally, $\sqrt{4 + x} \le 2 + \sqrt{x}$ gives the final inequality. □

A.3 CONSEQUENCES OF THE FULL-HORIZON CERTIFICATE

Proof of the low-loss bound (5). Theorem 3.1 gives

$$B_T \le B_f + \frac{C}{\sqrt{T}}\sqrt{B_T} + \frac{D}{2T}.$$

Let $x = B_T$, $a = C/\sqrt{T}$, and $d = D/(2T)$. For $u = \sqrt{x}$,

$$u^2 \le B_f + au + d, \qquad \text{so} \qquad \left(u - \frac{a}{2}\right)^2 \le B_f + d + \frac{a^2}{4}.$$

Taking the positive square root gives

$$u \le \frac{a}{2} + \sqrt{B_f + d + \frac{a^2}{4}}.$$

Hence

$$x \le B_f + d + a^2 + a\sqrt{B_f + d} \le B_f + a\sqrt{B_f} + \frac{3}{2}a^2 + \frac{3}{2}d,$$

where the last inequality uses $\sqrt{B_f + d} \le \sqrt{B_f} + \sqrt{d}$ and $a\sqrt{d} \le (a^2 + d)/2$. Substituting the definitions of a and d proves the claim. □

A.4 Smooth reweightings and binary edge

If $w \in [0,1]^T$ has density at least ρ, then its normalization $D_w(t) = w_t / \sum_s w_s$ satisfies $D_w(t) \le 1/(\rho T)$. Conversely, every distribution D on $[T]$ satisfying this upper bound is represented by the ρ-smooth reweighting $w_t = \rho T D(t)$. Thus our reweightings are the unnormalized form of the smooth distributions used by SmoothBoost (Servedio, 2003) and boosting-based hard-core constructions (Klivans & Servedio, 2003; Barak et al., 2009).

For a binary-valued h, let

$$\mathrm{err}_w(h) = \frac{\sum_t w_t \mathbf{1}\{h(x_t) \ne \sigma_t\}}{\sum_t w_t}.$$

Its normalized weighted correlation is $1 - 2\,\mathrm{err}_w(h)$, and $\mathrm{err}_w(-h) = 1 - \mathrm{err}_w(h)$. Hence

$$|1 - 2\,\mathrm{err}_w(h)| = 1 - 2\min\{\mathrm{err}_w(h), \mathrm{err}_w(-h)\}.$$

For a symmetric binary class, edge at least γ is therefore equivalent to the existence of a weak classifier with weighted error at most $(1-\gamma)/2$.

A.5 Strongly adaptive extension

We now state the exact interval guarantees summarized in Section 4. The construction first wraps each online learner so that its second-order regret bound holds on every interval.

Proposition A.3 (Second-order interval wrapper)**.** *Fix a horizon T. Suppose an online learner B, whenever started fresh, outputs $z_t \in [-1,1]$ and, for every comparator sequence $z^\star = (z_t^\star)_t$ in a fixed class and every $c_t \in [-2,2]$, satisfies*

$$\sum_{t=1}^{n} c_t(z_t^\star - z_t) \le a\sqrt{\sum_{t=1}^{n} c_t^2} + b.$$

Set $L_T = \log(4T)$ and $M_T = 2\lceil \log_2(2T) \rceil$. There is a wrapper $\mathsf{SA}(\mathsf{B})$ whose output $\widetilde{z}_t \in [-1,1]$ satisfies, simultaneously for every interval $I \subseteq [T]$ and comparator $z^\star$,

$$\sum_{t \in I} c_t(z_t^\star - \widetilde{z}_t) \le \alpha_T(a)\sqrt{\sum_{t \in I} c_t^2} + \beta_T(b), \tag{6}$$

where, for a universal constant C_0,

$$\alpha_T(a) = \sqrt{M_T}\big(a + C_0\sqrt{L_T}\big), \qquad \beta_T(b) = M_T\big(b + C_0 L_T\big).$$

The wrapper maintains at most $1 + \lceil \log_2 T \rceil$ active copies of B per round.

Apply Proposition A.3 separately to the weak-class learner and the scalar learners S, A, use their aggregate outputs in the same root rule, and feed the wrappers the same coefficients as in Algorithm 1. Call the result the *strongly adaptive Defensive Booster*. Let

$$a_H^{\mathrm{int}} = \alpha_T(a_{\mathcal{H}}), \quad b_H^{\mathrm{int}} = \beta_T(b_{\mathcal{H}}), \qquad a_{\mathrm{sc}}^{\mathrm{int}} = \alpha_T(4), \quad b_{\mathrm{sc}}^{\mathrm{int}} = \beta_T(8).$$

Theorem A.4 (Interval certificate). *For every adaptive binary sequence, the strongly adaptive Defensive Booster satisfies, simultaneously for every interval $I \subseteq [T]$,*

$$\sup_{h\in\mathcal{H}}\left|\sum_{t\in I} h(x_t)r_t\right| \le A_H^{\mathrm{int}}\sqrt{S_I} + B_H^{\mathrm{int}}, \qquad \left|\sum_{t\in I}\mu_t r_t\right| \le A_S^{\mathrm{int}}\sqrt{S_I} + B_S^{\mathrm{int}},$$

where $S_I = \sum_{t\in I} r_t^2$ and

$$A_H^{\mathrm{int}} = a_H^{\mathrm{int}} + a_{\mathrm{sc}}^{\mathrm{int}}, \quad B_H^{\mathrm{int}} = b_H^{\mathrm{int}} + b_{\mathrm{sc}}^{\mathrm{int}}, \qquad A_S^{\mathrm{int}} = 2a_{\mathrm{sc}}^{\mathrm{int}}, \quad B_S^{\mathrm{int}} = 2b_{\mathrm{sc}}^{\mathrm{int}}.$$

Corollary A.5 (Strongly adaptive boosting). *For an interval $I \subseteq [T]$, let $n = |I|$, $B_I = n^{-1}\sum_{t\in I}(Y_t - p_t)^2$, and $\rho_I = n^{-1}\sum_{t\in I}|Y_t - p_t|$. If $\rho_I > 0$, then*

$$\sup_{h\in\mathcal{H}} \frac{|\sum_{t\in I}|Y_t - p_t|\sigma_t h(x_t)|}{\sum_{t\in I}|Y_t - p_t|} \le \frac{A_H^{\mathrm{int}}\sqrt{nB_I} + B_H^{\mathrm{int}}/2}{n\rho_I}.$$

Simultaneously for every interval I:

(i) for every $f \in \mathrm{span}_\Lambda(\mathcal{H})$,

$$B_I \le \frac{1}{n}\sum_{t\in I}(Y_t - q_f(x_t))^2 + \frac{\Lambda A_H^{\mathrm{int}} + A_S^{\mathrm{int}}}{\sqrt{n}}\sqrt{B_I} + \frac{\Lambda B_H^{\mathrm{int}} + B_S^{\mathrm{int}}}{2n};$$

(ii) if every $w \in [0,1]^I$ of density at least ρ_0 has edge at least γ_0, then

$$B_I, \rho_I \le \max\left\{\rho_0, \frac{4(A_H^{\mathrm{int}})^2}{\gamma_0^2 n}, \frac{B_H^{\mathrm{int}}}{\gamma_0 n}\right\}.$$

The threshold classifier has error at most $2\rho_I$ on I.

The corollary follows by applying the full-horizon arguments on I with Theorem A.4. For fixed oracle constants and $\rho_0 = O(\varepsilon)$, target interval error ε requires $n = O(\log^2(T)/(\gamma_0^2\varepsilon))$. The interval and its comparator may be selected after observing the transcript; no assumption is made outside I.

Proof of Proposition A.3. Pad $[T]$ to the next power of two and let $\mathcal{J}$ be its canonical family of dyadic intervals. The family has fewer than $4T$ members, at most $1 + \lceil \log_2 T \rceil$ of which contain any round, and every interval $I \subseteq [T]$ is a disjoint union of at most M_T members of $\mathcal{J}$ (Daniely et al., 2015). Start one copy B_J at the left endpoint of each $J \in \mathcal{J}$ and run it only on J.

Aggregate the active copies with a second-order confidence-rated experts algorithm, treating membership in J as expert J's confidence. To see that the standard guarantee applies, let $q_{J,t}$ be the algorithm's weights on the active intervals and set $\widetilde{z}_t = \sum_J q_{J,t} z_{J,t}$. Map the linear gain to the loss

$$\ell_{J,t} = \frac{2 - c_t z_{J,t}}{4} \in [0,1].$$

The confidence-regret reduction and second-order excess-loss bound of Gaillard et al. (2014) give, for every $J \in \mathcal{J}$,

$$\sum_{t\in J} c_t(z_{J,t} - \widetilde{z}_t) \le C_0\left(\sqrt{L_T\sum_{t\in J} c_t^2} + L_T\right),$$

for a universal constant C_0. Here we assign a uniform prior to the fewer than $4T$ dyadic specialists. A geometric grid of learning rates, if needed, contributes only $O(\log\log T)$, which is absorbed by $C_0 L_T$. The loss range used in the second-order bound is valid because $|\ell_{J,t} - \sum_{J'} q_{J',t}\ell_{J',t}| \le |c_t|/2$.

Now partition an arbitrary I into $J_1, \dots, J_m \in \mathcal{J}$. On each block, insert the prediction of B_{J_j} between the comparator and the wrapper. The fresh-run guarantee and the preceding confidence-regret bound give

$$\begin{aligned}\sum_{t\in I} c_t(z_t^\star - \widetilde{z}_t) &\le \sum_{j=1}^m\left(a\sqrt{\sum_{t\in J_j} c_t^2} + b + C_0\sqrt{L_T\sum_{t\in J_j} c_t^2} + C_0 L_T\right)\\ &\le \sqrt{M_T}\left(a + C_0\sqrt{L_T}\right)\sqrt{\sum_{t\in I} c_t^2} + M_T\left(b + C_0 L_T\right),\end{aligned}$$

where the last step uses Cauchy–Schwarz and $m \le M_T$. Only the $1 + \lceil \log_2 T \rceil$ copies associated with intervals containing the current round are active. This proves (6).

Advance knowledge of T is not essential. Partition time into epochs $[2^j, 2^{j+1} - 1]$ and run the fixed-horizon construction afresh in each epoch. Any interval up to time T meets at most $1 + \lceil \log_2 T \rceil$ epochs. Summing the fixed-horizon bounds over those pieces and applying Cauchy–Schwarz adds a factor $\sqrt{1 + \lceil \log_2 T \rceil}$ to the coefficient of the second-order term and a factor $1 + \lceil \log_2 T \rceil$ to the additive term. At any time only the wrapper for the current epoch is active. □

Proof of Theorem A.4. Fix an interval I. The root sign property holds separately on every round, so the aggregated auditor gain is nonpositive on I. Interval regret for the wrapped A against its two endpoint comparators therefore gives

$$\sum_{t \in I} z_{H,t}, \quad \sum_{t \in I} z_{S,t} \le a_{\mathrm{sc}}^{\mathrm{int}} \sqrt{S_I} + b_{\mathrm{sc}}^{\mathrm{int}}.$$

For every $h \in \mathcal{H}$, interval regret for the wrapped weak-class oracle then gives

$$\sum_{t \in I} h(x_t) r_t \le \sum_{t \in I} z_{H,t} + a_H^{\mathrm{int}} \sqrt{S_I} + b_H^{\mathrm{int}} \le A_H^{\mathrm{int}} \sqrt{S_I} + B_H^{\mathrm{int}}.$$

Symmetry of $\mathcal{H}$ gives the absolute value. Similarly, for $\theta \in \{-1, 1\}$, interval regret for the wrapped S gives

$$\sum_{t \in I} \theta \mu_t r_t \le \sum_{t \in I} z_{S,t} + a_{\mathrm{sc}}^{\mathrm{int}} \sqrt{\sum_{t \in I} \mu_t^2 r_t^2} + b_{\mathrm{sc}}^{\mathrm{int}} \le A_S^{\mathrm{int}} \sqrt{S_I} + B_S^{\mathrm{int}}.$$

Taking both signs proves the self-orthogonality inequality. □

B Separation: the guarantees are incomparable

The Brier/span and weak-to-strong guarantees do not imply one another. Proposition B.1 gives a transcript on which every reweighting has positive weak-class edge, yet every real-valued score induced by the span has constant squared loss. The construction exploits a basic difference between the guarantees: squared loss depends on the numerical values of a score, whereas weak-to-strong boosting can exploit its sign. Proposition B.2 gives the converse separation: a small subset of uninformative rounds defeats the smooth weak-learning condition even though a span comparator has small squared loss.

Proposition B.1 (Separation). *For every $\gamma \in (0, 1)$ and $\eta > 0$, there are a number $\delta \in [\gamma, \min\{\gamma + \eta, 1\})$, a symmetric binary-valued class $\mathcal{H}$, and a transcript $(x_t, Y_t)_{t \le T}$ such that:*

- *(i) every reweighting $w \in [0, 1]^T$ with $\sum_t w_t > 0$ has $\mathrm{edge}_{\mathcal{H}}(w) \ge \delta$; in particular the (ρ, γ)-smooth weak-learning condition holds for every $\rho > 0$;*
- *(ii) no single hypothesis is correct on every round, but a uniform average of hypotheses in $\mathcal{H}$ has positive signed margin on every round and therefore classifies the transcript perfectly;*
- *(iii) every affine score $q_f(x) = (1 + f(x))/2$ induced by the span of $\mathcal{H}$, with no range or norm constraint, has average squared loss at least $(1 - \delta)^2/(8(1 + \delta^2))$.*

Proof. Choose an integer k large enough that $2/k < \min\{\eta, 1 - \gamma\}$, and set

$$r = \left\lceil \frac{(1 + \gamma)k}{2} \right\rceil, \qquad \delta = \frac{2r - k}{k}.$$

Then $k/2 < r < k$ and $\gamma \le \delta < \min\{\gamma + \eta, 1\}$. Let $h_1, \ldots, h_k$ be binary-valued hypotheses and include their negations in $\mathcal{H}$. All labels equal one, so $\sigma_t = 1$ on every round.

The transcript has two equally large parts. On every context in the first part, set $h_j(x_t) = 1$ for all j. The second part contains one context for each r-element subset $A \subseteq [k]$, and on that context set

$$h_j(x_A) = \begin{cases} 1, & j \in A, \\ -1, & j \notin A. \end{cases}$$

Repeat the all-positive context $\binom{k}{r}$ times so that the two parts have equal size. At every context,

$$\frac{1}{k}\sum_{j=1}^{k}\sigma_t h_j(x_t) \geq \delta.$$

Consequently, for every reweighting w with positive mass,

$$\frac{1}{k}\sum_{j=1}^{k}\frac{\sum_{t=1}^{T} w_t\sigma_t h_j(x_t)}{\sum_{t=1}^{T} w_t} \geq \delta.$$

Some h_j must therefore have weighted edge at least δ, proving (i). The uniform average $k^{-1}\sum_j h_j$ equals 1 on the first part and δ on the second, so its sign is always correct. On the other hand, every h_j equals -1 on a positive fraction of the second part because $r < k$. This proves (ii).

It remains to minimize squared loss over the span. Write $f = \sum_{j=1}^{k}\alpha_j h_j$. The transcript and its squared-loss objective are invariant under permutations of the k coordinates. Averaging the coefficient vector over all permutations and using convexity therefore cannot increase loss. Hence an optimum has $\alpha_1 = \cdots = \alpha_k = b/k$ for some $b \in \mathbb{R}$. Its signed score equals b on the first part and δb on the second, so its Brier loss is

$$\psi(b) = \frac{(1-b)^2 + (1-\delta b)^2}{8}.$$

This is minimized at $b^\star = (1+\delta)/(1+\delta^2)$, where it equals $(1-\delta)^2/(8(1+\delta^2))$. This proves (iii). □

The separation also survives clipping span scores to the probability range when the coefficient norm is bounded. Fix $\Lambda \geq 1$, write $f = \sum_j \alpha_j h_j$ with $\sum_j |\alpha_j| \leq \Lambda$, and clip f to $[-1,1]$ before converting it to a probability. On the second half of the construction, the average of f over all r-subsets is $\delta\sum_j \alpha_j \leq \delta\Lambda$. For every $u \in [-\Lambda, \Lambda]$,

$$\mathrm{clip}_{[-1,1]}(u) \leq \frac{2u + \Lambda - 1}{\Lambda + 1}.$$

If $\delta \leq 1/(2\Lambda)$, the average clipped signed score on this half is therefore at most $\Lambda/(\Lambda+1)$. Jensen's inequality shows that these rounds have average Brier loss at least $1/(4(\Lambda+1)^2)$, and hence the full transcript has average Brier loss at least $1/(8(\Lambda+1)^2)$. In particular, clipping does not remove the separation for any fixed coefficient-norm budget. Moreover, perfect clipped prediction would require $f(x_A) \geq 1$ for every r-subset A; averaging these inequalities gives $\delta\sum_j \alpha_j \geq 1$, and therefore $\sum_j |\alpha_j| \geq 1/\delta$.

As $\delta \to 0$, the span's best squared loss approaches $1/8$, while Corollary 3.4 forces the Defensive Booster's Brier score and classification error to vanish as T grows. Thus a Brier/span-regret guarantee alone can leave constant squared loss on transcripts where the smooth weak-learning condition forces near-perfect prediction.

The reverse implication fails for a different reason: a small set of rounds can support a zero-edge reweighting while contributing little to average squared loss.

Proposition B.2 (Converse separation)**.** *For every T and every even $m \in \{2,\ldots,T\}$, there is a symmetric binary-valued class $\mathcal{H} = \{h_1, -h_1, h_2, -h_2\}$ and a transcript $(x_t, Y_t)_{t\leq T}$ such that:*

(i) the smooth weak-learning condition fails for every $\rho \leq m/T$ and every $\gamma > 0$;

(ii) the span comparator $f = (h_1+h_2)/2 \in \mathrm{span}_1(\mathcal{H})$ induces the score $q_f = (1+f)/2$ with average squared loss $T^{-1}\sum_t (Y_t - q_f(x_t))^2 = m/(4T)$.

Proof. Choose distinct contexts $x_1,\ldots,x_T$, arbitrary binary labels, and let $R \subseteq \{1,\ldots,T\}$ have size m. Split R into equal parts R_1 and R_2. Outside R, set $h_1(x_t) = h_2(x_t) = \sigma_t$. On R_1,

set $(\sigma_t h_1(x_t), \sigma_t h_2(x_t)) = (1, -1)$; on R_2, reverse these two values. Consider the reweighting $w_t = \mathbf{1}\{t \in R\}$. It has density m/T, and

$$\sum_{t=1}^{T} w_t \sigma_t h_1(x_t) = \sum_{t=1}^{T} w_t \sigma_t h_2(x_t) = 0$$

and the same holds for their negations, so $\mathrm{edge}_{\mathcal{H}}(w) = 0$. Hence the (ρ, γ)-smooth weak-learning condition fails for every $\rho \le m/T$ and every $\gamma > 0$.

The comparator $f = (h_1 + h_2)/2$ equals σ_t off R and zero on R. It therefore predicts perfectly off R and induces probability $1/2$ on R, so each round in R contributes squared loss $1/4$. Its average squared loss is therefore $m/(4T)$. □

The random-label mixture experiment in Section 5 instantiates the second separation qualitatively: the randomly labeled rounds support a smooth reweighting with small weak-class edge, while the structured rounds retain an informative least-squares span score.

C Additional related work

This section supplies broader historical and technical context for the comparisons in Section 6.

Online gradient and classification boosting. Beyond the online gradient boosting result most directly comparable to our span guarantee, Hu et al. (2017) study gradient boosting on i.i.d. streams and extend their analysis to adversarial streams under a stronger edge assumption, while Hazan & Singh (2021) use a multiplicative weak learner to obtain regret to a convex hull in online convex optimization. Practical online bagging and boosting goes back to Oza & Russell (2001). The closest smooth classification predecessor is the online SmoothBoost algorithm of Chen et al. (2012); Beygelzimer et al. (2015b) give Online BBM and matching lower bounds in their weak online-learning model. In that model, the optimal sample-complexity dependence for error ε is $1/(\gamma^2\varepsilon)$ up to logarithmic and excess-loss terms.

The oracle assumptions differ. Online SmoothBoost and Online BBM assume that the online learner's predictions have a fixed positive advantage over random guessing on every admissible stream, with smooth importance weights in the former case, up to an excess-loss term. The multiplicative agnostic oracle of Brukhim et al. (2020) obtains a fixed fraction of the best correlation in $\mathcal{H}$ and is boosted to regret against the best $h \in \mathcal{H}$. Our primitive instead has no-regret under linear losses: it need not have absolute edge, and it uses $\mathcal{H}$ as directions that are aggregated into forecasts competing with $\mathrm{span}(\mathcal{H})$. Positive edge enters separately as an ex-post property of smooth transcript reweightings.

Boosting as loss optimization. The interpretation of boosting as functional gradient descent was developed by Mason et al. (1999). Mukherjee et al. (2013) show that AdaBoost converges to the infimum empirical exponential loss over additive combinations of weak hypotheses without assuming weak learnability or a finite minimizer. This is analogous to our unconditional span comparison, but their guarantee concerns batch optimization of exponential margin loss, whereas ours is an online pathwise Brier guarantee for probabilities. Exponential loss does not by itself produce calibrated probabilities; Niculescu-Mizil & Caruana (2005) document poor squared error and cross-entropy for uncalibrated boosted outputs.

Strong adaptivity. Strongly adaptive online learning asks for low regret on every contiguous interval. Geometric-cover reductions obtain this guarantee from a standard online learner with $O(\log T)$ active copies (Daniely et al., 2015); second-order confidence bounds preserve dependence on local gradient energy (Gaillard et al., 2014; Cutkosky, 2020). AdaBoost.OL is parameter-free rather than strongly adaptive in this interval sense. Appendix A.5 applies the strongly adaptive machinery inside all three auditors, producing both interval span regret and local hard-core witnesses.

Smooth boosting and hard-core sets. Smooth distributions are central to SmoothBoost (Servedio, 2003), and Gavinsky (2003) develops smooth adaptive boosting in the agnostic setting. The

minimax formulation of boosting goes back to Freund & Schapire (1996). The connection between boosting and hard-core construction begins with Impagliazzo (1995) and was made algorithmic by Klivans & Servedio (2003) and Barak et al. (2009). The mistake weights in Theorem 3.3 are an online transcript analogue of these smooth hard-core distributions.

Multicalibration, multiaccuracy, and loss minimization. Multicalibration was introduced by Hébert-Johnson et al. (2018); multiaccuracy was isolated as a black-box correction criterion by Kim et al. (2019). Outcome indistinguishability and omniprediction turn stronger prediction certificates into simultaneous downstream loss guarantees (Dwork et al., 2021; Gopalan et al., 2022; 2023). Globus-Harris et al. (2023) characterize when batch multicalibration boosts squared-loss regression to Bayes optimality, and Kearns et al. (2025) use multiaccuracy together with self-orthogonality for loss minimization. Stronger calibration conditions also yield hard-core measures (Casacuberta et al., 2024; 2025). Our weak-to-strong argument requires only multiaccuracy. This weaker condition is important online because adversarial sequential calibration cannot generally be bounded at the $O(\sqrt{T})$ scale achieved here (Qiao & Valiant, 2021; Dagan et al., 2025; Collina et al., 2026).

Defensive forecasting. Defensive forecasting chooses probabilities that prevent continuous "skeptic" strategies from increasing their capital (Vovk et al., 2005b;a). Vovk (2007) also handles continuous "second-guessing" experts whose advice depends on the current forecast; the affine test in Algorithm 1 has this form. Many online calibration and multicalibration algorithms admit a defensive-forecasting interpretation (Gupta et al., 2022; Bastani et al., 2022; Noarov & Roth, 2023; Noarov et al., 2025; Garg et al., 2024; Ghuge et al., 2025; Perdomo & Recht, 2025; Hu et al., 2026). Farina & Perdomo (2026) give generic reductions from online multicalibration to a no-regret learner plus an expected variational-inequality solver and recover traditional defensive forecasting as special cases. Our exact affine root is a deterministic one-dimensional instance of that framework.

D Additional experimental details and results

This appendix section gives the complete protocol and additional results for the binary prediction experiments. The synthetic streams separately examine the two guarantees: Brier loss and an offline least-squares span comparator measure competition with the span, while randomized classification error $T^{-1}\sum_t |Y_t - p_t|$ measures weak-to-strong performance and equals the density of the forecaster's mistake weighting. The real streams evaluate forecasting and classification performance on examples processed in their recorded order. Appendices E.1 and F report the regression and strongly adaptive experiments, respectively.

D.1 Protocol and implementation

Algorithms. The main comparison includes eight methods. Defensive implements the Defensive Booster (Algorithm 1), including the two class-independent scalar adaptive-OGD states in Definition A.1. For the real and distribution-shift experiments, we additionally report Adaptive Def., the strongly adaptive Defensive Booster from Section 4; it maintains one weak learner at each active dyadic scale. Section F gives its implementation details and local diagnostics.

The two unboosted controls each maintain a single learner. Unboosted reg. performs online squared-loss regression over the same weak class, using online linear optimization on the squared-loss gradient. Unboosted cls. runs the online binary classifier used as the weak learner by Online BBM, AdaBoost.OL, and OSBoost; it isolates the benefit of boosting from that of the base classification algorithm.

The four ensemble baselines each maintain 100 weak learners. OGB is the online gradient boosting algorithm of Beygelzimer et al. (2015a), specialized to one-dimensional signed squared loss, equivalently Brier loss up to a factor of four, and run with 100 boosting stages and stage step $\eta = (\log N)/N$, as suggested by their theoretical discussion. In the learner-count sweep, the $N = 1$ endpoint uses the admissible step $\eta = 1$. Online BBM is the boost-by-majority algorithm of Beygelzimer et al. (2015b), the optimal-rate binary online classification booster; we run its importance-weighted version with 100 weak learners and the target advantage described below. Online BBM outputs hard binary predictions, so its Brier score in our tables is the Brier score of the induced $0/1$ probability forecast. AdaBoost.OL is the adaptive logistic-loss booster from the

same paper. We implement the importance-weighted version used in its experiments: projected online gradient descent learns each weak learner's coefficient, and exponential weights aggregate the hard predictions of the partial ensembles. The algorithm's probability of predicting one is used as its forecast. Thus its randomized-error score is the expected $0/1$ error of the original randomized classifier, while its Brier score evaluates that probability forecast directly. AdaBoost.OL requires no target-edge parameter. OSBOOST is the online SmoothBoost algorithm of Chen et al. (2012) with its online-convex-programming combiner and 100 weak learners, following the experimental convention in that paper. We use the target-selection rule below and importance-weighted updates as in their experiments. We include OSBoost because its smooth reweighting mechanism is especially close to the smooth-distribution mechanism analyzed here.

The Online BBM/AdaBoost.OL paper and the OSBoost paper define γ as the amount by which weak classification error improves on $1/2$; equivalently, a binary weak prediction has signed correlation 2γ. On a synthetic stream with a known guaranteed advantage, we supply that value rather than tune it from the observed results. Thus Online BBM and OSBoost use $\gamma = .08$ on binary aggregation, whose guaranteed correlation edge is .16, and $\gamma = .1$ on the other streams. This convention keeps the baselines' classification parameter distinct from the correlation edge in our theorem.

Finally, BRIER AGGREGATOR runs OGB, Online BBM, AdaBoost.OL, and OSBoost in parallel. Before observing Y_t, it predicts the weighted average of their four probabilities and, after observing Y_t, multiplies each weight by $\exp\{-(Y_t - p_{t,i})^2/2\}$. Since Brier loss is $1/2$-exp-concave on $[0, 1]$, its cumulative Brier loss is at most that of the best constituent plus $2 \log 4$: for each binary outcome y, $p \mapsto \exp(-(y - p)^2/2)$ is concave, so the standard exponential-weights potential argument applies to the weighted-average forecast. This baseline maintains all 400 constituent weak learners.

To place all outputs on the same probability scale, we interpret each method's signed score $s_t \in [-1, 1]$ as the probability $p_t = (1+s_t)/2$. OGB projects its aggregate score to $[-1, 1]$, OSBoost uses its simplex-weighted vote, AdaBoost.OL uses its exponential-weights randomization probability, and UNBOOSTED CLS. and Online BBM return $s_t \in \{-1, 1\}$; we apply no post-hoc calibration. Thus the reported Brier loss evaluates the probability induced directly by each online algorithm's output.

For each algorithm family, we use one predeclared tuning rule rather than optimizing parameters against the observed performance of each stream. In DEFENSIVE, both scalar states use Definition A.1 with $V_0 = 4$; no scalar learning rate is tuned. The linear-loss oracles used by DEFENSIVE, UNBOOSTED REG., and every OGB stage are second-order. For a finite class with d base hypotheses, we run entropy-FTRL over its symmetric closure with

$$\eta_t = \min\left\{.25, \sqrt{\frac{\log(2d)}{4 + \sum_{s<t} c_s^2}}\right\},$$

where c_s is the scalar multiplying the weak prediction in the round-s linear gain. For the Euclidean unit ball, we use projected adaptive gradient ascent with step $.5/\sqrt{4 + \sum_{s<t} \|c_s x_s\|_2^2}$. These choices give the coefficient-energy regret bounds required by the theory. The finite classification learners used by UNBOOSTED CLS., Online BBM, AdaBoost.OL, and OSBoost instead share the horizon-aware Hedge scale $\min\{.5, \sqrt{8 \log(d)/T}\}$; their linear-class counterparts share the same weighted projected-perceptron update with a $1/\sqrt{t}$ step scale.

All synthetic results average 20 random seeds with $T = 3000$ rounds, and tables report mean $\pm$ standard error. The real datasets have fixed chronological order and deterministic algorithm updates, so we report one run. We never shuffle a real stream: each example is predicted in its recorded order before it is used for the update. The `experiments/` directory contains the generators, loaders, algorithm implementations, and commands used for every reported result. The synthetic streams are original controlled constructions rather than reproductions of prior benchmarks.

Runtime. Let C_H be the cost of one prediction/update for the online learner over the weak class. Each unboosted control and the Defensive Booster costs $C_H + O(1)$ per round. OGB, Online BBM, AdaBoost.OL, and OSBoost each cost $NC_H + O(N)$ with N weak learners; OSBoost additionally pays an $O(N \log N)$ simplex projection when the OCP combiner updates. We use $N = 100$ for all ensemble baselines in the main comparison, and additionally evaluate each ensemble baseline

with $N \in \{1, 5, 20, 100\}$ learners. The Brier aggregator runs all four ensembles and therefore costs $4NC_H + O(N \log N)$ and maintains $4N$ weak learners.

D.2 Controlled synthetic streams

Synthetic streams. We use five streams to separate two questions: whether every smooth reweighting has a weak rule with positive edge, and whether the span contains an informative squared-loss comparator. The planted-decoy and binary aggregation streams satisfy the smooth weak-learning condition by construction. The three linear streams instead test span competition with noisy labels or weak signal.

For the two finite-class streams, an algorithm uses a context only through the vector of weak-rule values, so we generate that vector directly. Round t is represented by $(h_1(x_t), \dots, h_d(x_t))$, and the linear-loss oracle uses the symmetric closure of the coordinate rules. For the other three streams, the weak class is the Euclidean unit ball $\mathcal{H} = \{x \mapsto \langle u, x\rangle : \|u\|_2 \le 1\}$.

In *planted decoy*, σ_t is uniform on $\{-1, 1\}$, the useful rule is $h_1(x_t) = s_t\sigma_t$ for an independent uniform $s_t \in \{.12, 1\}$, and the other 199 rule scores are independent uniform signs. Thus the 200 base rules (and their symmetric closure) contain a sign-perfect rule with edge at least .12 under every reweighting.

In *binary aggregation*, the displayed weak class consists of 100 opposite pairs $\{\pm h_j\}_{j=1}^{100}$, and signed labels are balanced and randomly ordered. One latent orientation from each pair is designated useful. On half the rounds all 100 useful orientations equal the signed label. On the remaining rounds, we cycle through 100 binary patterns: each pattern has 58 useful orientations correct and 42 incorrect, and every useful orientation occupies each position equally often. Independent sign flips and a column permutation hide the useful orientation in each pair; we also randomly permute the rounds. The algorithms receive only the resulting vector of 200 weak-rule values. On every round,

$$\frac{1}{100}\sum_{j=1}^{100} \sigma_t h_j(x_t) \ge .16.$$

Here the sum uses the latent useful orientation from each pair. After multiplying by any nonzero weights and averaging, at least one displayed rule has edge .16. No individual rule is perfect, and the average of all 200 displayed rules is identically zero, whereas the hidden average above has positive margin on every round.

Adding the opposite rules, applying sign flips, and permuting the columns do not change the span. The same symmetry therefore yields an exact span obstruction. Averaging any coefficient vector over cyclic shifts cannot increase squared loss, so an optimal fixed span score assigns every latent useful orientation the same coefficient. Its signed score is b on the first half of the stream and $.16b$ on the second half. Minimizing the resulting Brier loss gives

$$\min_{b\in\mathbb{R}} \frac{(1-b)^2 + (1-.16b)^2}{8} = \frac{(1-.16)^2}{8(1+.16^2)} = .0860.$$

Thus the stream is a finite cyclic version of the binary separation in Proposition B.1: aggregation classifies perfectly, but every fixed affine span score has constant Brier loss.

For the three linear streams, draw $x_t \sim N(0, I_d)$ and normalize it to unit Euclidean norm; draw and normalize a fixed $\beta \sim N(0, I_d)$. In *linear span*, $d = 40$ and $\sigma_t = \mathrm{sign}(\langle \beta, x_t\rangle + \xi_t)$ for $\xi_t \sim N(0, .02^2)$. In *random-label mixture*, $d = 30$ and the same model uses noise $N(0, .05^2)$, but independently on 35% of rounds its label is replaced by a uniform random sign. The reweighting supported on the replaced labels has density close to .35, and because those labels are independent of the contexts, its weak-class edge tends to zero as T grows. The structured rounds still admit an informative linear score. In *random labels*, $d = 30$ and every signed label is an independent uniform sign. All draws are independent except where the construction explicitly shares σ_t or β.

Offline span diagnostic. For each realized synthetic transcript, let $\widehat{\beta} \in \arg\min_\beta \sum_t (\sigma_t - \langle \beta, x_t\rangle)^2$. The induced real-valued prediction is $q_{\widehat{\beta}}(x) = (1 + \langle \widehat{\beta}, x\rangle)/2$, whose average squared

loss is exactly

$$\frac{1}{T}\sum_{t=1}^{T}(Y_t - q_{\widehat{\beta}}(x_t))^2 = \frac{1}{4T}\sum_{t=1}^{T}(\sigma_t - \langle\widehat{\beta}, x_t\rangle)^2.$$

We report this loss and a norm Λ_{LS} witnessing membership in the comparator class of Theorem 3.1: $\|\widehat{\beta}\|_1$ for a finite coordinate class and $\|\widehat{\beta}\|_2$ for the Euclidean linear ball. The score is not clipped, so it is an unrestricted span comparator covered by Theorem 3.1. Because $\widehat{\beta}$ is fit after observing the full transcript, it is a diagnostic benchmark rather than an online algorithm.

Figure 4 reports randomized classification error on the two streams satisfying the smooth weak-learning condition. Figure 5 reports Brier loss on the three linear streams. Tables 2 and 3 give the corresponding span benchmarks and complete numerical results.

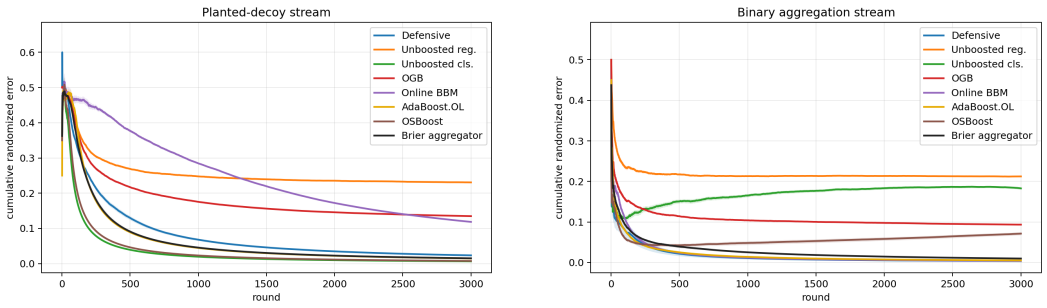

Figure 4: Randomized classification error on the streams satisfying the smooth weak-learning condition. Left: on planted decoy, one base rule has the correct sign on every round but is hidden among 199 decoys; the unboosted classifier and OSBoost identify it fastest, and the Defensive Booster also reaches low error. Right: on binary aggregation, no individual rule is perfect and the average of all displayed rules is zero, but a hidden choice of one orientation from each opposite pair has positive margin. The Defensive Booster and the three weak-to-strong ensembles all attain low randomized error; the Defensive Booster uses one weak learner.

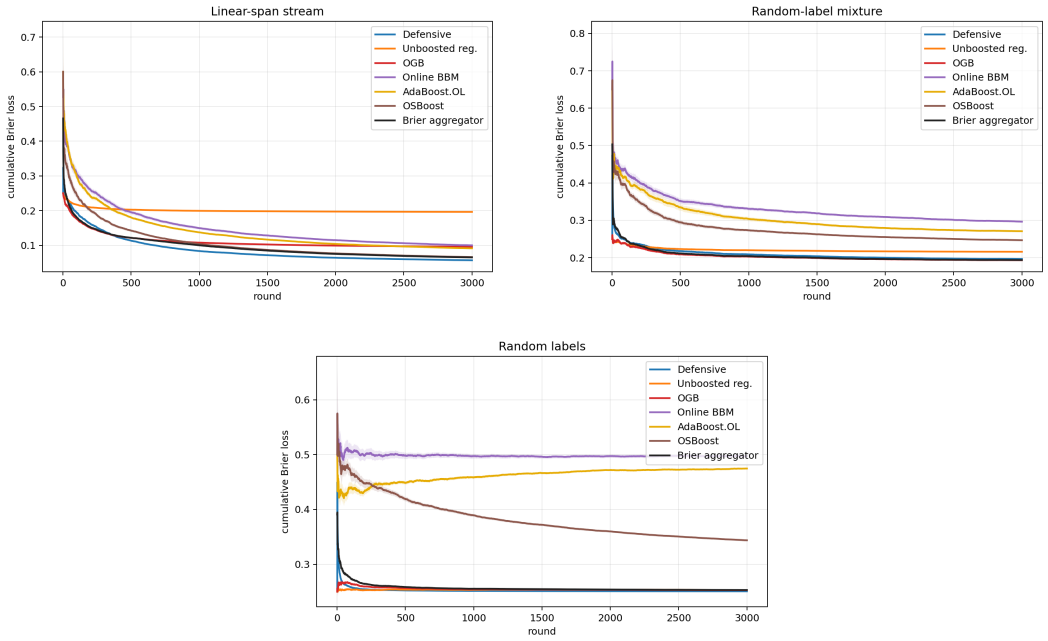

Figure 5: Brier loss on the three linear streams. Top left: on linear span, the span comparator is informative even though the smooth weak-learning condition can fail on near-margin examples. Top right: on random-label mixture, the structured rounds retain an informative span comparator; OGB, unboosted regression, and the Defensive Booster outperform the classification-boosting baselines. Bottom: on random labels, the Defensive Booster, unboosted regression, and OGB remain near the $p = 1/2$ Brier baseline. The hard-label unboosted classifier is omitted from these plots for scale and reported in Table 3.

Stream	LS span squared loss	Λ_{LS}
Planted decoy	$.0887 \pm .0004$	$2.982 \pm .020$
Binary aggregation	$.0860 \pm .0000$	$1.131 \pm .000$
Linear span	$.0900 \pm .0003$	$5.084 \pm .017$
Random-label mixture	$.1847 \pm .0007$	$2.794 \pm .018$
Random labels	$.2472 \pm .0001$	$.576 \pm .013$

Table 2: Offline least-squares span diagnostics, averaged over the same 20 seeds used for the online results. The score is not clipped. The reported Λ_{LS} is the ℓ_1 norm of the fitted coefficient vector for a finite class or the ℓ_2 norm for the Euclidean linear class. Hence each row is an actual $\mathrm{span}_{\Lambda_{\mathrm{LS}}}(\mathcal{H})$ comparator covered by Theorem 3.1.

Stream	Algorithm	0/1	Brier	Rand. err.
Planted decoy	Defensive	$.0166 \pm .0008$	$.0119 \pm .0005$	$.0235 \pm .0010$
Planted decoy	Unboosted reg.	$.0072 \pm .0004$	$.0999 \pm .0004$	$.2306 \pm .0009$
Planted decoy	Unboosted cls.	$.0065 \pm .0003$	$.0065 \pm .0003$	$.0065 \pm .0003$
Planted decoy	OGB	$.0162 \pm .0005$	$.0390 \pm .0002$	$.1348 \pm .0005$
Planted decoy	BBM	$.1185 \pm .0008$	$.1185 \pm .0008$	$.1185 \pm .0008$
Planted decoy	AdaBoost.OL	$.0147 \pm .0004$	$.0135 \pm .0004$	$.0148 \pm .0004$
Planted decoy	OSBoost	$.0073 \pm .0003$	$.0071 \pm .0002$	$.0077 \pm .0003$
Planted decoy	Brier agg.	$.0096 \pm .0005$	$.0069 \pm .0003$	$.0152 \pm .0006$
Binary aggregation	Defensive	$.0026 \pm .0006$	$.0018 \pm .0005$	$.0036 \pm .0009$
Binary aggregation	Unboosted reg.	$.1026 \pm .0013$	$.0944 \pm .0001$	$.2124 \pm .0002$
Binary aggregation	Unboosted cls.	$.1829 \pm .0028$	$.1829 \pm .0028$	$.1829 \pm .0028$
Binary aggregation	OGB	$.0331 \pm .0009$	$.0338 \pm .0002$	$.0935 \pm .0003$
Binary aggregation	BBM	$.0041 \pm .0002$	$.0041 \pm .0002$	$.0041 \pm .0002$
Binary aggregation	AdaBoost.OL	$.0042 \pm .0004$	$.0033 \pm .0002$	$.0052 \pm .0005$
Binary aggregation	OSBoost	$.0068 \pm .0007$	$.0192 \pm .0014$	$.0713 \pm .0037$
Binary aggregation	Brier agg.	$.0025 \pm .0002$	$.0025 \pm .0001$	$.0099 \pm .0007$
Linear span	Defensive	$.0744 \pm .0007$	$.0570 \pm .0004$	$.1120 \pm .0008$
Linear span	Unboosted reg.	$.0952 \pm .0010$	$.1964 \pm .0001$	$.4403 \pm .0002$
Linear span	Unboosted cls.	$.1233 \pm .0015$	$.1233 \pm .0015$	$.1233 \pm .0015$
Linear span	OGB	$.0866 \pm .0010$	$.0960 \pm .0003$	$.2511 \pm .0006$
Linear span	BBM	$.1002 \pm .0014$	$.1002 \pm .0014$	$.1002 \pm .0014$
Linear span	AdaBoost.OL	$.0966 \pm .0011$	$.0915 \pm .0009$	$.0971 \pm .0010$
Linear span	OSBoost	$.0849 \pm .0011$	$.0663 \pm .0008$	$.0953 \pm .0011$
Linear span	Brier agg.	$.0815 \pm .0011$	$.0653 \pm .0008$	$.1327 \pm .0028$
Random-label mixture	Defensive	$.2667 \pm .0017$	$.1965 \pm .0007$	$.3890 \pm .0015$
Random-label mixture	Unboosted reg.	$.2671 \pm .0017$	$.2158 \pm .0003$	$.4576 \pm .0003$
Random-label mixture	Unboosted cls.	$.2789 \pm .0018$	$.2789 \pm .0018$	$.2789 \pm .0018$
Random-label mixture	OGB	$.2656 \pm .0013$	$.1933 \pm .0007$	$.3859 \pm .0013$
Random-label mixture	BBM	$.2963 \pm .0021$	$.2963 \pm .0021$	$.2963 \pm .0021$
Random-label mixture	AdaBoost.OL	$.2758 \pm .0022$	$.2708 \pm .0021$	$.2757 \pm .0022$
Random-label mixture	OSBoost	$.3260 \pm .0024$	$.2467 \pm .0012$	$.3655 \pm .0018$
Random-label mixture	Brier agg.	$.2658 \pm .0013$	$.1937 \pm .0007$	$.3859 \pm .0014$
Random labels	Defensive	$.4980 \pm .0020$	$.2506 \pm .0000$	$.4999 \pm .0001$
Random labels	Unboosted reg.	$.4970 \pm .0016$	$.2529 \pm .0001$	$.4995 \pm .0002$
Random labels	Unboosted cls.	$.4978 \pm .0015$	$.4978 \pm .0015$	$.4978 \pm .0015$
Random labels	OGB	$.4968 \pm .0019$	$.2527 \pm .0001$	$.4995 \pm .0002$
Random labels	BBM	$.4971 \pm .0025$	$.4971 \pm .0025$	$.4971 \pm .0025$
Random labels	AdaBoost.OL	$.4983 \pm .0020$	$.4745 \pm .0014$	$.4995 \pm .0019$
Random labels	OSBoost	$.4994 \pm .0024$	$.3437 \pm .0015$	$.4977 \pm .0016$
Random labels	Brier agg.	$.4966 \pm .0018$	$.2530 \pm .0001$	$.4994 \pm .0002$

Table 3: Average online performance over 20 random seeds, reported as mean $\pm$ standard error. "Rand. err." is $T^{-1}\sum_t |Y_t - p_t|$, the error of the randomized classifier induced by the probability forecast. UNBOOSTED CLS. and BBM output hard labels, so their Brier and randomized-error entries equal their $0/1$ error. BRIER AGG. combines the forecasts of OGB, BBM, AdaBoost.OL, and OSBoost before observing the current label, then updates its weights after the label is revealed; it therefore maintains 400 weak learners. AdaBoost.OL's Brier entry scores its probability of predicting one, while its randomized-error entry is the expected $0/1$ loss of its randomized classifier. Entries are rounded to four decimals.

Online BBM is designed to output a hard majority prediction. To check whether its Brier results are merely an artifact of that convention, we also score the normalized raw vote $(1+N^{-1}\sum_i h_{t,i}(x_t))/2$ as a probability. This diagnostic is not the output analyzed by the Online BBM theorem. Figure 6 shows that the normalized vote lowers planted-decoy Brier loss from .1185 to .0838, but remains far above the best forecasting methods. On binary aggregation, the hard output has Brier loss and randomized error .0041, while the normalized vote has Brier loss .0520 and randomized error .1562. Thus a softer output helps on planted decoy but hurts substantially on binary aggregation; it does not account for the main comparisons.

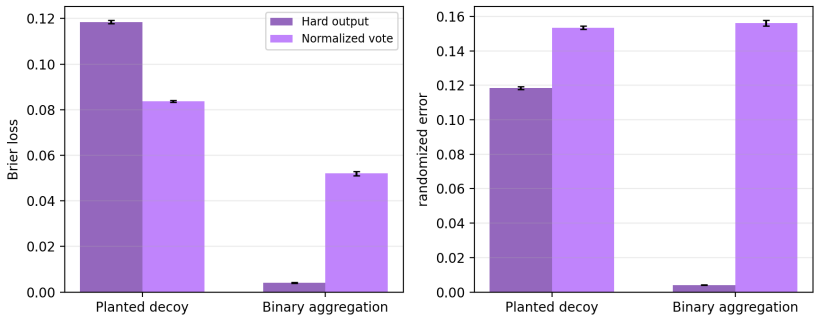

Figure 6: Online BBM's specified hard output versus its normalized raw vote, averaged over 20 seeds. The vote is scored directly as a probability, with no post-hoc calibration. It improves Brier loss on planted decoy but worsens randomized error there; on binary aggregation, the specified hard output is substantially better under both metrics.

Algorithm	Weak learners	Seconds per stream	Microseconds per round
Unboosted reg.	1	.032	11
Unboosted cls.	1	.030	10
Defensive	1	.046	15
OGB	100	2.959	986
BBM	100	1.478	493
AdaBoost.OL	100	2.292	764
OSBoost	100	1.753	584
Brier aggregator	400	8.482	2827

Table 4: Runtime summary for the implementation used in the experiments. Wall-clock times are averages over all stream/seed runs, each with $T = 3000$ rounds. The exact constants are implementation-dependent, but the difference in maintained learners is structural: both unboosted controls and the Defensive Booster maintain one online learner, whereas the ensemble baselines maintain many weak learners in parallel. The Brier aggregator's time is the sum of its four constituent ensembles because all must be run.

Sensitivity to baseline parameters. The main experiments use one setting for each algorithm family on every stream. Figure 7 varies, one at a time, the OGB stage step, the classification learner's Hedge rate, and the target-edge parameter on the binary aggregation stream. Each value is .25, .5, 1, 2, or 4 times the reported setting, and each point averages 10 seeds. OGB improves steadily with its stage step but has higher Brier loss than the Defensive Booster at every tested value. Online BBM is stable across both sweeps, and AdaBoost.OL has slightly lower randomized error than the Defensive Booster at the smallest tested learning rate. OSBoost is substantially more sensitive to both its classification learning rate and target edge. Thus the main comparison does not depend on a single narrow baseline setting, although the relative ordering of the lowest-error classification methods can change.

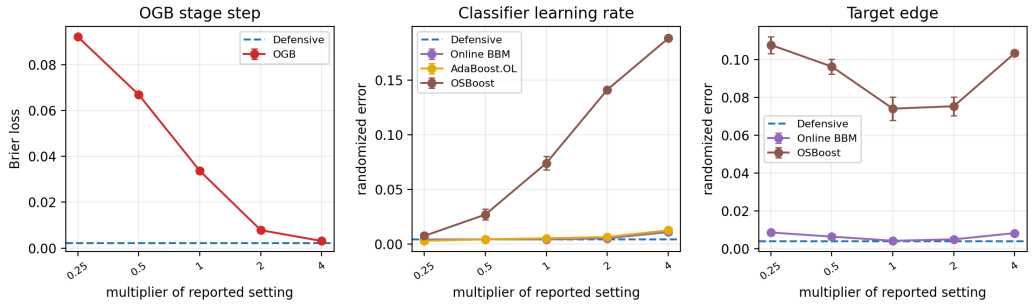

Figure 7: One-at-a-time parameter sensitivity on the binary aggregation stream. The dashed blue line is the parameter-free Defensive Booster. The middle and right panels show randomized classification error; the left panel shows Brier loss. Error bars are standard errors over 10 seeds. A target-edge multiplier above one deliberately overstates the edge guaranteed by the construction and is included as a misspecification check.

Figure 8 repeats the same sixteen-fold sweep on Electricity and Occupancy, the two real streams on which the Defensive Booster has the largest advantage. It retains the lowest Brier loss under every tested setting. This check does not tune the reported results: the main tables continue to use multiplier one for every dataset.

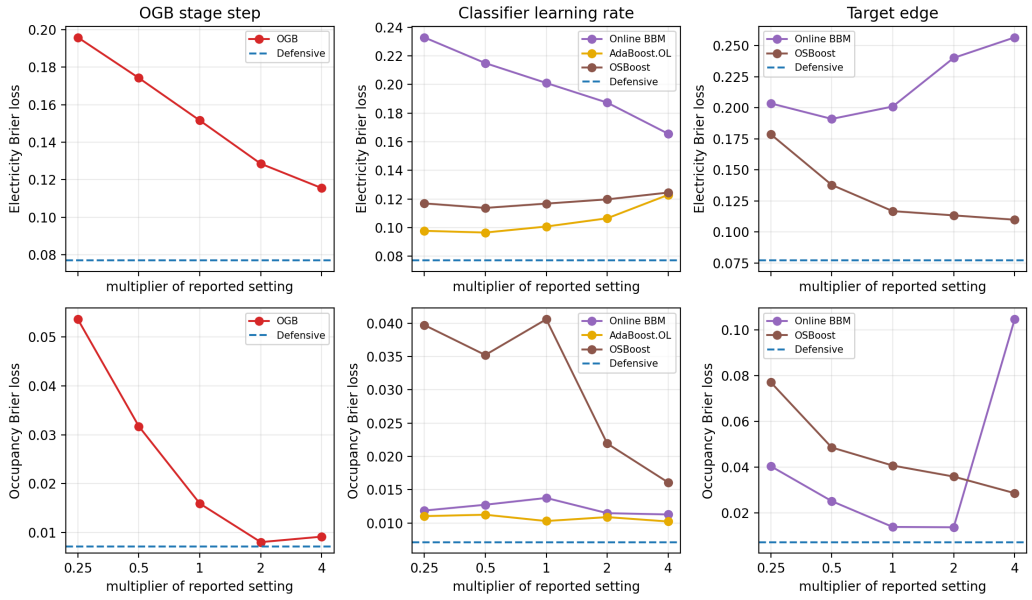

Figure 8: One-at-a-time parameter sensitivity on the complete Electricity (top) and Occupancy (bottom) streams. Multipliers range from .25 to 4. The dashed blue line is the parameter-free Defensive Booster. Every panel shows final Brier loss; lower is better. AdaBoost.OL has no target-edge parameter and therefore appears only in the middle column.

Takeaways. The planted-decoy stream tests identification within a large weak class, but does not require boosting: one base rule already has the correct sign on every round. Accordingly, the unboosted classifier and OSBoost have the smallest errors, and the Defensive Booster also attains low Brier and randomized error. Unboosted regression has low threshold error but poor Brier and randomized error, illustrating why probability forecasting is stricter than threshold accuracy. The unrestricted least-squares span score has loss .0887, whereas the Defensive Booster reaches .0119; this weak-to-strong behavior is not explained by span fitting.

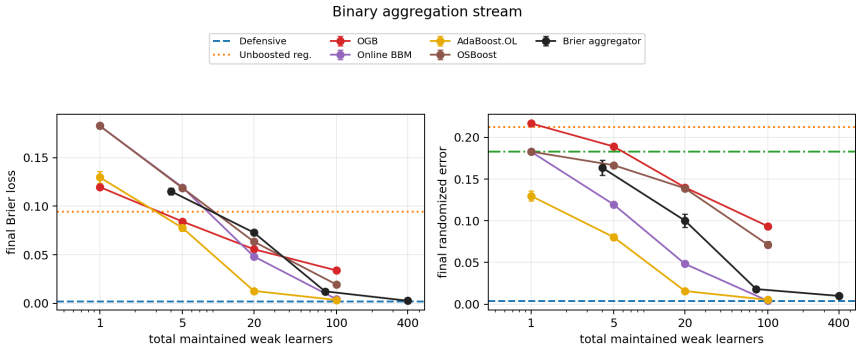

Figure 9: Prediction quality as a function of the number of maintained weak learners on the binary aggregation stream. OGB, Online BBM, AdaBoost.OL, and OSBoost use $N \in \{1, 5, 20, 100\}$ learners; the Brier aggregator runs all four ensembles and is plotted at $4N$. The Defensive Booster and unboosted controls each maintain one learner.

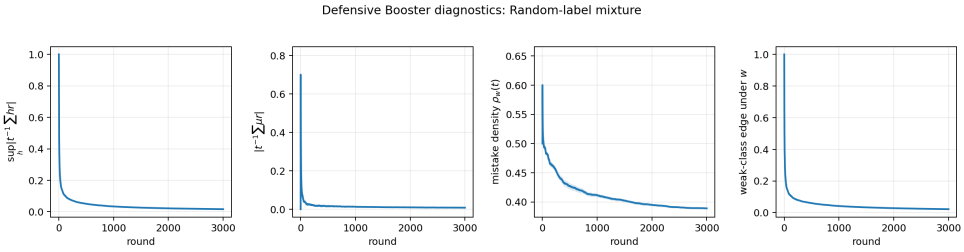

Figure 10: Certificate diagnostics on the random-label mixture, averaged over 20 seeds. The first two panels show cumulative multiaccuracy and self-orthogonality correlations divided by the current round; the final two show the density and weak-class edge of the mistake weighting $w_t = |Y_t - p_t|$. The weighting retains nontrivial density while its edge decays.

The binary aggregation stream requires genuine aggregation: no individual binary rule is perfect, and averaging all displayed rules gives zero. A hidden choice of one orientation from each pair nevertheless classifies every round correctly. The unboosted classifier has error .1829, while the Defensive Booster reaches Brier loss .0018 and randomized error .0036 using one weak learner, despite the .0860 loss floor for fixed affine span scores. At $N = 100$, all three weak-to-strong ensembles also reach low hard error, but each has higher Brier loss and randomized error than the Defensive Booster. The ensemble-size comparison in Figure 9 shows how these values compare with each ensemble as its number of weak learners grows.

The three linear streams in Figure 5 examine Brier loss when the smooth weak-learning condition need not hold. On linear span, the offline span score has loss .0900, and the Defensive Booster attains .0570. On random-label mixture, the randomly labeled rounds provide a smooth reweighting with small weak-class edge, while the structured rounds retain an informative span comparator. OGB, the unboosted regressor, and the Defensive Booster have substantially lower Brier loss than the unboosted classifier and the classification-boosting baselines. OSBoost and Online BBM remain more competitive in randomized error than in Brier loss because randomized error depends linearly on their signed margins. Figure 10 examines the random-label mixture more closely: the Defensive Booster's multiaccuracy and self-orthogonality errors and the weak-class edge under its mistake weighting all decay, although the weighting retains nontrivial density.

Finally, on random labels, no method has a real signal; the Defensive Booster, unboosted regressor, and OGB stay near the $p = 1/2$ Brier-loss baseline, while the unboosted classifier and the classification boosters have larger Brier loss. The weak-class edge under the Defensive Booster's mistake weighting remains small.

Under these globally fixed tuning rules, the classification boosters are strongest on the binary aggregation stream, while OGB is strongest on the random-label mixture. The Defensive Booster is competitive with the better family in both comparisons. Table 4 shows that it does so while maintaining one online weak learner rather than an ensemble of 100 learners.

D.3 Naturally ordered real streams

Data and preprocessing. We evaluate four public binary prediction streams in their recorded order. UCI Bank Marketing predicts whether a client subscribes to a term deposit (Moro et al., 2014); we use the date order supplied by the full dataset. The MOA Electricity stream (MOA, 2011) predicts price movement in the New South Wales electricity market; the data originate in Harries's electricity-pricing study (Harries, 1999). The MOA Airlines stream predicts whether a flight is delayed (MOA, 2011). UCI Occupancy Detection (Candanedo & Feldheim, 2016) predicts whether an office is occupied from minute-level sensor measurements, which we merge by recorded timestamp. We do not shuffle any dataset: on each round the algorithm receives the next context, predicts, and then observes its label.

We use a 128-dimensional deterministic signed-hash representation and the Euclidean unit-ball weak class for all four datasets. Each row includes a bias feature and is normalized to unit norm. Categorical values are hashed as indicators; numeric values are standardized using unlabeled covariates from earlier rounds, clipped to five running standard deviations, and then hashed. The current numeric value is transformed using the mean and variance of that feature among preceding contexts and is incorporated into those statistics only afterward. Thus no feature or label from a future round enters the current representation. For Bank Marketing we drop call duration, which is unavailable before the outcome. Occupancy uses contemporaneous temperature, humidity, light, and CO_2 measurements, the derived humidity ratio, and cyclic encodings of the recorded time and weekday. The Bank, Electricity, Airlines, and Occupancy runs use all 41,188, 45,312, 539,383, and 20,560 examples, respectively.

Figure 11 plots cumulative average Brier loss over each stream. Table 5 reports the final Brier, deterministic, and randomized classification errors, together with runtime.

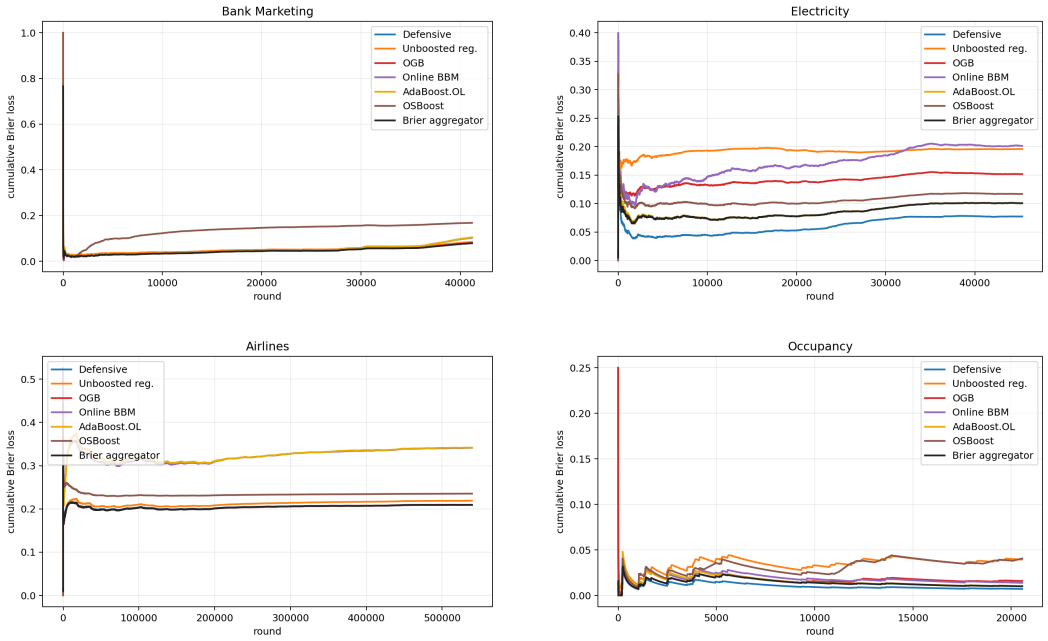

Figure 11: Cumulative average Brier loss on the real-data streams. Top left: on Bank Marketing, OGB and the Defensive Booster are close. Top right: on Electricity, the Defensive Booster has substantially lower Brier loss than all six plotted baselines. Bottom left: on Airlines, the Defensive Booster, OGB, and the Brier aggregator are nearly indistinguishable. Bottom right: on Occupancy, the Defensive Booster has the lowest Brier loss. Each curve is the average loss incurred up to that point while processing the dataset in recorded order. The hard-label unboosted classifier is omitted for scale and reported in Table 5.

Dataset	Algorithm	0/1	Brier	Base	Rand. err.	μs/round
Bank	Defensive	.103	.080	.100	.159	14
Bank	Adaptive Def.	.103	.081	.100	.159	86
Bank	Unboosted reg.	.103	.085	.100	.212	10
Bank	Unboosted cls.	.109	.109	.100	.109	8
Bank	OGB	.102	.079	.100	.157	803
Bank	BBM	.103	.103	.100	.103	300
Bank	AdaBoost.OL	.106	.105	.100	.106	522
Bank	OSBoost	.131	.168	.100	.387	356
Bank	Brier agg.	.102	.079	.100	.148	1981
Electricity	Defensive	.108	.077	.244	.154	15
Electricity	Adaptive Def.	.085	.064	.244	.129	86
Electricity	Unboosted reg.	.273	.196	.244	.422	11
Electricity	Unboosted cls.	.367	.367	.244	.367	8
Electricity	OGB	.219	.152	.244	.310	807
Electricity	BBM	.201	.201	.244	.201	329
Electricity	AdaBoost.OL	.102	.101	.244	.102	526
Electricity	OSBoost	.111	.117	.244	.280	349
Electricity	Brier agg.	.102	.101	.244	.102	2011
Airlines	Defensive	.332	.209	.247	.419	16
Airlines	Adaptive Def.	.324	.207	.247	.413	89
Airlines	Unboosted reg.	.348	.219	.247	.451	11
Airlines	Unboosted cls.	.390	.390	.247	.390	10
Airlines	OGB	.332	.209	.247	.419	826
Airlines	BBM	.341	.341	.247	.341	352
Airlines	AdaBoost.OL	.342	.342	.247	.342	543
Airlines	OSBoost	.370	.235	.247	.480	362
Airlines	Brier agg.	.332	.209	.247	.419	2082
Occupancy	Defensive	.009	.007	.178	.014	14
Occupancy	Adaptive Def.	.007	.007	.178	.013	85
Occupancy	Unboosted reg.	.052	.040	.178	.120	9
Occupancy	Unboosted cls.	.096	.096	.178	.096	8
Occupancy	OGB	.022	.016	.178	.033	838
Occupancy	BBM	.014	.014	.178	.014	288
Occupancy	AdaBoost.OL	.011	.010	.178	.011	536
Occupancy	OSBoost	.015	.041	.178	.088	382
Occupancy	Brier agg.	.012	.010	.178	.016	2044

Table 5: Average online performance on the real-data streams. "Base" is the Brier loss of the best constant probability forecast on the evaluated dataset. UNBOOSTED CLS. and BBM report the Brier score of their hard binary predictions. BRIER AGG. runs all four ensemble baselines. Runtime is reported for the same implementation and machine as Table 4. The absolute times are implementation-dependent. The basic Defensive Booster maintains one weak-class learner, ADAPTIVE DEF. maintains one at each active dyadic scale (20 for Airlines, 17 for Bank and Electricity, and 16 for Occupancy), each ensemble baseline maintains 100, and the Brier aggregator maintains 400.

Among the eight methods in the main comparison, the Defensive Booster has the lowest Brier loss on Electricity and Occupancy. On Occupancy its Brier loss .007 is less than half OGB's .016; it also has the lowest deterministic error. AdaBoost.OL has the lowest randomized error on Occupancy and the lowest two classification errors on Electricity. These distinctions are consistent with the algorithms' objectives: the Defensive Booster is designed to forecast probabilities, while AdaBoost.OL directly optimizes classification. The Brier gains are not explained merely by maintaining fewer learners: both unboosted controls are substantially worse on Electricity and Occupancy. The Brier aggregator is best on Bank by .0010 over the Defensive Booster. On Airlines, OGB has the numerically smallest Brier loss, but it, the aggregator, and the Defensive Booster differ by less than $6 \cdot 10^{-5}$. The unboosted regressor is also competitive on these two streams, while the classification methods are worse on Brier loss. On Electricity and Occupancy, however, the Defensive Booster's Brier loss is respectively 61% and 82% lower than the unboosted regressor's. The Brier aggregator maintains 400 weak learners. On the four real streams, the $N = 100$ ensembles take 20–60$\times$ as much wall-clock time per round as the Defensive Booster, which maintains one weak learner; the Brier aggregator takes about 130–150$\times$ as much. The separate Brier and classification columns

matter here: a boosted margin method can classify accurately without producing the most accurate probability forecasts.

E Extension to bounded real-valued outcomes

It is evident that the Defensive Booster, as it is defined, does not require that the outcomes be binary; it can be applied essentially without modification in a regression setting with bounded scalar labels. We will now briefly state and discuss this natural extension: in a nutshell, the Defensive Booster's span guarantee is satisfied in the exact same way as in the binary setting. Then, we will evaluate this extension on three chronological regression streams — in these experiments, the Defensive Booster has 17–29% lower normalized mean squared error than 100-stage online gradient boosting while maintaining one weak-class learner; OGB takes 65–70× as much wall-clock time per round in our implementation.

To begin, suppose that $Y_t \in [0,1]$ and that $p_t \in [0,1]$ is interpreted as a prediction of the bounded outcome. Retain the affine encoding

$$\sigma_t = 2Y_t - 1, \qquad \mu_t = 2p_t - 1, \qquad r_t = 2(Y_t - p_t).$$

The Defensive Booster is unchanged; also note that outcomes in any other fixed bounded interval reduce to this setting by affine rescaling. It is now easy to see that the span guarantee remains the same.

Proposition E.1 (Certificate and span guarantee for bounded outcomes)**.** *For every adaptive sequence with $Y_t \in [0,1]$, the Defensive Booster satisfies the multiaccuracy and self-orthogonality guarantees of Theorem 2.3, with*

$$S_T = 4\sum_{t=1}^{T}(Y_t - p_t)^2.$$

In particular, writing $L_{2,T} = T^{-1}\sum_t (Y_t - p_t)^2$, for every $f \in \mathrm{span}_\Lambda(\mathcal{H})$,

$$L_{2,T} \le \frac{1}{T}\sum_{t=1}^{T}(Y_t - q_f(x_t))^2 + \frac{\Lambda A_H + A_S}{\sqrt{T}}\sqrt{L_{2,T}} + \frac{\Lambda B_H + B_S}{2T}.$$

The low-loss conclusion of Corollary 3.2 also holds with B_T and B_f interpreted as the corresponding average squared errors.

Proof. Lemma 2.2 already permits every $\sigma_t \in [-1,1]$. The proof of Theorem 2.3 therefore applies unchanged. The proof of Theorem 3.1 then uses only that certificate and the same squared-loss expansion, so it also applies unchanged. □

E.1 Regression experiments with bounded outcomes

We test the Defensive Booster beyond binary outcomes on three public regression datasets, processed in timestamp order. *Appliance energy* records household appliance use every ten minutes together with indoor and outdoor sensor measurements (Candanedo, 2017). *Bike demand* records hourly Capital Bikeshare rentals with calendar and weather variables (Fanaee-T, 2013). *Interstate traffic* records hourly westbound I-94 traffic volume with calendar and weather variables (Hogue, 2019).

We normalize each target by a fixed upper bound R: 2000 Wh for Appliance Energy, 2000 rentals per hour for Bike Demand, and 10000 vehicles per hour for Interstate Traffic. We map a target $Y_t \in [0,R]$ to $Y_t/R \in [0,1]$ and map a normalized forecast p_t back to Rp_t. Thus

$$(Y_t/R - p_t)^2 = (Y_t - Rp_t)^2/R^2,$$

so root mean squared error in the original units is R times the square root of normalized MSE. Every observed target lies in its stated interval, so this normalization clips no outcome.

We use the first 10% of each stream as a common chronological initialization prefix and report losses on the remaining 90%. Thereafter, each algorithm receives the current context, predicts, observes

the target, and updates. Numeric features are standardized using statistics from strictly earlier rows. Each context contains calendar variables and the available sensor or weather measurements. Bike and Traffic additionally use targets observed exactly one hour, one day, and one week earlier when those timestamps exist; Appliance Energy uses lags of ten minutes, one hour, and one day. We omit simultaneous light consumption and two random decoy columns from Appliance Energy. We omit the casual and registered rental counts from Bike because they sum to the target, and collapse duplicate weather reports at a Traffic timestamp.

All learned methods (except the past-outcome mean baseline) receive the same 128-dimensional signed feature-hashed context, rescaled to have Euclidean norm at most one, and use the same unit-ball linear weak class. Algorithmic hyperparameters are fixed across datasets. The Defensive Booster maintains one second-order linear oracle; OGB maintains $N = 100$ such oracles and uses the stage step $\eta = (\log N)/N$ from the binary experiments. Two controls use either one unboosted squared-loss learner or the mean of targets observed before the current round. Figure 12 plots cumulative mean squared error on the normalized targets, and Table 6 reports final normalized MSE and root mean squared error in the original units.

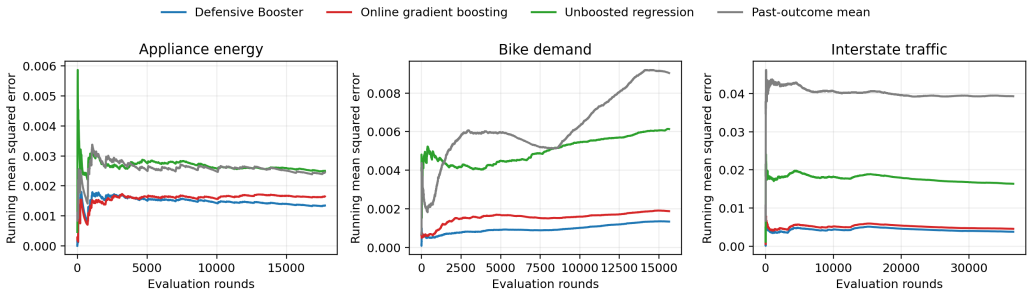

Figure 12: Cumulative mean squared error after the common chronological initialization prefix. The Defensive Booster and unboosted control each maintain one weak-class learner, whereas online gradient boosting maintains 100. The Defensive Booster has the lowest final loss on all three streams.

		Normalized MSE		Raw-unit RMSE		OGB time
Dataset	Rounds	Defensive	OGB	Defensive	OGB	/ Defensive
Appliance energy	17,762	**.0013443**	.0016461	**73.330**	81.145	64.7×
Bike demand	15,642	**.0013440**	.0018824	**73.322**	86.773	65.9×
Interstate traffic	36,518	**.0038040**	.0045762	**616.769**	676.473	69.6×

Table 6: Final regression performance after the common chronological initialization prefix; lower error is better. Raw-unit RMSE is measured in Wh, rentals per hour, and vehicles per hour, respectively. The final column is the ratio of OGB's wall-clock time per round to the Defensive Booster's on the same machine. Absolute runtimes depend on the implementation, whereas OGB's 100 maintained weak-class learners versus the Defensive Booster's one is an algorithmic difference.

The Defensive Booster reduces normalized MSE relative to OGB by 18% on Appliance Energy, 29% on Bike Demand, and 17% on Interstate Traffic. On Bike and Traffic, both controls are substantially worse, so the improvement does not come merely from predicting the running mean or from applying the shared weak learner once. Thus the empirical advantage extends beyond binary outcomes: on each stream, the Defensive Booster obtains lower squared error than the 100-learner OGB ensemble while maintaining one weak-class learner.

F Strongly adaptive experiments

We next compare the basic Defensive Booster with its strongly adaptive variant from Section 4. The implementation uses the canonical dyadic interval family and the Adapt-ML-Prod second-order aggregation rule with the sleeping-expert confidence reduction of Gaillard et al. (2014). The weak-class and scalar states are vectorized across scales, which reduces implementation overhead without

changing their updates. This variant introduces no dataset-specific parameter: the same second-order weak oracle and scalar routines are used at every scale.

The ADAPTIVE DEF. rows in Table 5 report its full-stream performance. On Electricity, strong adaptivity lowers Brier loss from .0772 to .0644, deterministic error from .1077 to .0851, and randomized error from .1538 to .1289. It also lowers Airlines Brier loss from .2094 to .2066 and Occupancy Brier loss from .0071 to .0069; on Bank it increases Brier loss from .0800 to .0807. The adaptive implementation takes 85–89 microseconds per round, about six times the basic forecaster but still substantially less than the 100-learner ensembles in Table 5. Figure 13 compares the methods' trailing-window losses and shows when these full-stream improvements occur.

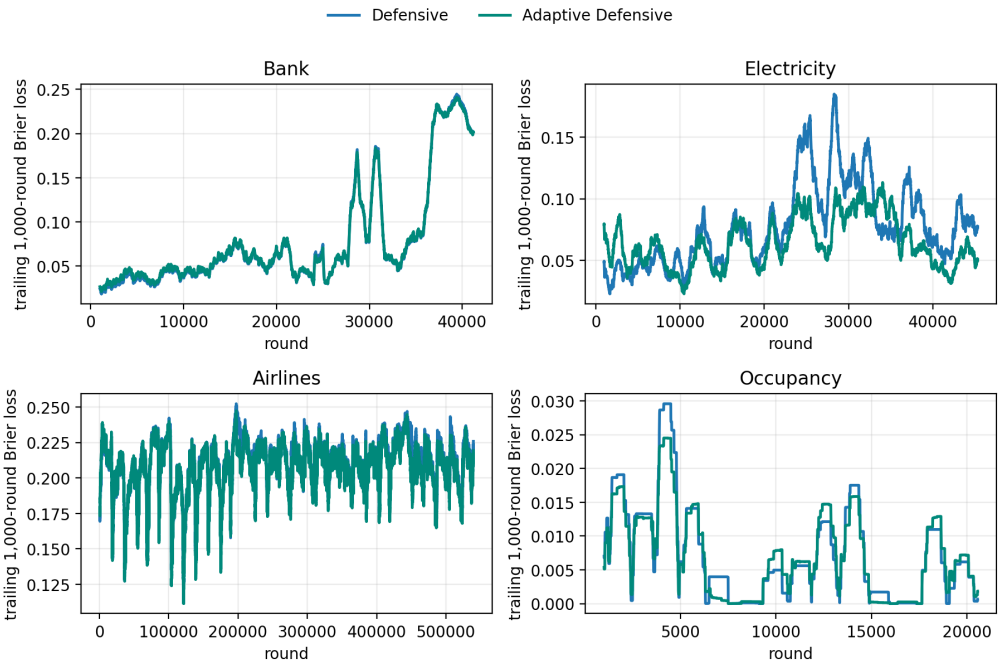

Figure 13: Local performance of the basic and strongly adaptive Defensive Boosters on all four real streams. Each curve is the trailing 1,000-round Brier loss; the same window is fixed for every dataset. The adaptive variant tracks the basic forecaster closely on Bank and Airlines, while on Electricity it improves substantially during the later high-loss portions of the stream. Occupancy is mixed locally but favors the adaptive variant in full-stream loss. Table 5 reports the corresponding full-stream averages.

Controlled drift benchmark. To isolate adaptation to distribution shift, we additionally use the INSECTS optical-sensor benchmark of Souza et al. (2020). Each example is an optical-sensor recording of mosquito flight. The benchmark orders these examples using a hidden temperature variable to produce known abrupt, incremental-gradual, and recurring drift patterns. We use all five balanced variants and preserve each released order. Because our setting is binary, we fix one target for the entire benchmark: recognize *Aedes albopictus* (either sex) versus *Aedes aegypti* or *Culex quinquefasciatus*. Each of the six source classes has equal frequency, so the binary target has positive rate $1/3$. The 33 numeric signal features receive the same prefix-only standardization and row normalization as the other real streams. Neither algorithm is given the temperature, the drift type, or the change points. Table 7 reports full-stream performance for all five released orderings.

Drift pattern	Rounds	Brier loss		0/1 error	
		Basic	Adaptive	Basic	Adaptive
Abrupt	52,848	.1307	**.1198**	.1905	**.1697**
Incremental-gradual	24,150	.0905	**.0877**	.1321	**.1235**
Incremental-abrupt recurring	79,986	.0871	**.0813**	.1231	**.1148**
Incremental recurring	79,986	.0829	**.0769**	.1151	**.1064**
Incremental	57,018	**.1700**	.1711	**.2535**	.2547

Table 7: Average online performance on five controlled-drift INSECTS streams; lower is better. "Basic" is the Defensive Booster, which maintains one weak learner; "Adaptive" is its strongly adaptive variant. The binary task and all algorithmic choices are fixed across rows.

The adaptive variant lowers both metrics on the four streams that combine abrupt, gradual, or recurring shifts. On the continuously incremental stream, the methods differ by at most .0012. The adaptive implementation takes 83–85 microseconds per round. On the four original real streams, Table 5 shows that the adaptive method is slower than the basic forecaster but remains 3–10× faster than the 100-learner ensembles. Figure 14 shows the local Brier losses around the published change points for two representative streams.

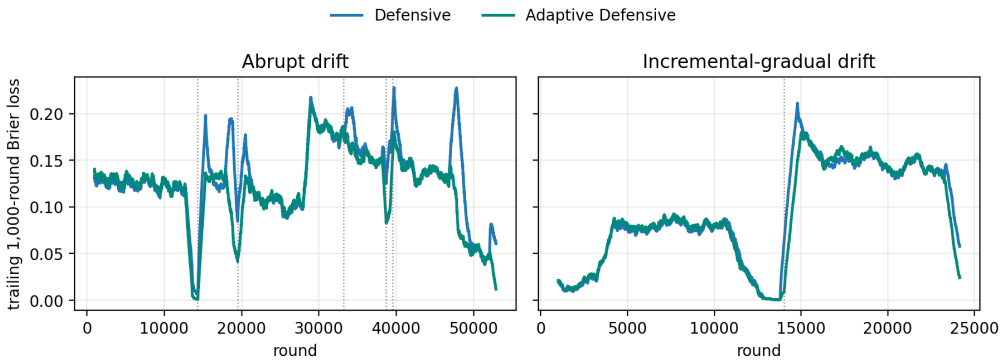

Figure 14: Local Brier loss on the abrupt and incremental-gradual INSECTS streams. Curves are trailing 1,000-round averages; dotted lines mark the change points published with the benchmark. The adaptive forecaster reduces several of the largest post-shift loss spikes.

The interval guarantee also produces local hard-core witnesses. Figure 15 examines these witnesses on the abrupt INSECTS stream. For each selected endpoint t and dyadic length L, it computes the mistake weighting on the trailing interval $I = [t - L + 1, t]$ and reports both quantities that define its hard-core quality: density and normalized weak-class edge.

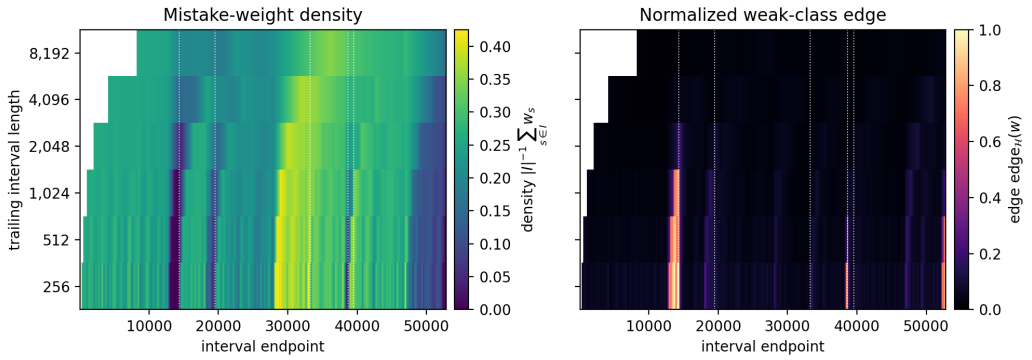

Figure 15: Evolution of local hard-core witnesses for the strongly adaptive Defensive Booster on the abrupt INSECTS stream. Each pixel represents the mistake weights $w_s = |Y_s - p_s|$ on one trailing interval $I = [t - L + 1, t]$: the horizontal axis is its endpoint t, and the vertical axis is its dyadic length L. The left panel gives the density $|I|^{-1} \sum_{s \in I} w_s$; the right gives $\mathrm{edge}_{\mathcal{H}}(w)$. Bright regions on the left paired with dark regions on the right are smooth, low-edge local hard-core witnesses. Dotted lines mark the five published abrupt change points; white regions precede the first complete interval at a given scale.