Trustworthy Online Conformal Prediction by Super Learner Ensembling

Yuhan (Victoria) Nian[1*], Jiaming Liu[1*] and Meng Li[1*]

[1*]Department of Statistics, Rice University, Houston, TX, USA.

*Corresponding author(s). E-mail(s): vn23@rice.edu; jl259@rice.edu; meng@rice.edu;

Abstract

Reliable uncertainty quantification is essential in streaming and nonstationary settings. Online conformal prediction (OCP) guarantees distribution-free coverage, but its efficiency is limited when based on a single model. We extend OCP with the Super Learner (SL), a meta-learning ensemble that combines heterogeneous predictors under constrained weights. Compared with Conformal Online Model Aggregation (COMA), our approach consistently yields narrower confidence intervals while maintaining validity. Experiments on synthetic jump–diffusion data and the `ELEC2` dataset show that: (i) the Super Learner outperforms individual base learners and COMA, (ii) when included as an expert it dominates COMA ensembles, and (iii) all methods preserve coverage near the nominal level. These results highlight meta-learning as a practical strategy for improving efficiency in online conformal prediction.

Keywords: Online Conformal Prediction, Super Learner, Model Aggregation

1 Introduction

Predictive modeling, such as time series forecasting, online recommendation, or adaptive medical monitoring, requires not only accurate predictions, but also reliable uncertainty quantification. A prediction is only actionable if it is accompanied by trustworthy confidence intervals that remain valid under distributional shifts and streaming data. In recent years, *conformal prediction* has gained attention for its ability to provide distribution-free prediction sets with finite-sample guarantees [1]. However, most

conformal methods assume data exchangeability and are designed for batch settings, limiting their applicability in real-time or nonstationary contexts.

To address this, **online conformal prediction (OCP)** extends conformal methods to the sequential setting. In particular, Angelopoulos et al. [2] proposed an online algorithm that updates prediction quantiles using a *decaying step size* mechanism. This approach maintains approximate marginal coverage while adapting to gradual changes in the data distribution. The update rule is lightweight, theoretically grounded, and suitable for real-time applications where recalibration must occur on-the-fly.

While OCP methods provide valid prediction sets, they often rely on a single underlying model, which can lead to suboptimal efficiency—i.e., unnecessarily wide prediction intervals—if the model is poorly calibrated or mismatched to the data.

In response to this limitation, we propose a method that integrates online conformal prediction (OCP) with the Super Learner ensemble (SL) [3], a meta-learning algorithm that optimally combines predictions from a library of models to minimize error. In our setting, each base learner is paired with its own online conformal module, producing a valid prediction interval. These intervals are then aggregated through a convex meta-learner that learns weights to minimize interval width while preserving coverage. Our key insight is that by optimally combining the strengths of multiple models, one can reduce instability and overly conservative behavior from individual learners, leading to prediction intervals that are narrower while still preserving conformal validity.

As a benchmark, we compare our approach to **Conformal Online Model Aggregation (COMA)** [4], which uses exponential-weights aggregation but does not explicitly exploit the flexibility of SL. Our framework is simple, modular, and general: it integrates seamlessly into online learning pipelines, preserves the robustness of conformal prediction, and consistently shrinks CI widths through adaptive ensemble weighting.

1.1 Related Work

Reliable uncertainty quantification has long been a central challenge in statistics and machine learning, particularly in settings where predictions must remain valid under distributional uncertainty. Conformal prediction (CP), first introduced by Vovk, Gammerman, and Shafer [1], addresses this challenge by constructing predictive intervals with rigorous coverage guarantees under minimal assumptions. By leveraging nonconformity scores, CP ensures finite-sample validity while adapting to the observed data.

Much of the classical CP literature assumes exchangeable data, limiting its applicability to sequential or streaming settings. To address this, Gibbs and Candès [5] introduced online conformal prediction with fixed-step updates, and Angelopoulos et al. [2] later proposed decaying step sizes that improve calibration under distributional shift. These works highlight the importance of dynamically adjusting conformal procedures to preserve validity and efficiency in nonstationary environments.

A complementary direction focuses on aggregation. Gasparin and Ramdas [4] proposed Conformal Online Model Aggregation (COMA), which treats multiple candidate forecasters as experts and adaptively reweights them using algorithms such as Hedge [6] and AdaHedge [7]. By combining predictors, COMA improves efficiency compared

to relying on any single model, while maintaining conformal guarantees. However, these expert-weighting strategies originate from online learning and do not exploit the optimality properties of statistical stacking methods.

In parallel, the statistical literature has developed powerful ensemble approaches for batch learning, most notably the Super Learner (SL) [3]. SL constructs an optimally weighted convex combination of candidate models via cross-validation and enjoys oracle properties under mild conditions. Despite its success in offline prediction, its integration into online conformal prediction remains unexplored. Our work addresses this gap by embedding SL into the COMA framework and directly comparing convex meta-learning with Hedge-style weighting. This connection bridges statistical stacking and online conformal methods, showing how meta-learning can further improve interval efficiency while preserving validity guarantees.

1.2 Our Contributions

This work advances online conformal prediction (OCP) and model aggregation along four dimensions.

1. We extend the OCP framework proposed by Angelopoulos et al. [2] to integrating a constrained SL meta-optimizer, enabling ensemble learning with multiple base predictors in the online setting.
2. We provide a systematic comparison with Conformal Online Model Aggregation (COMA) [4], highlighting the differences in aggregation strategies and evaluating their relative performance.
3. We show both theoretically and empirically that incorporating the SL yields narrower conformal prediction intervals while maintaining valid coverage, thereby improving predictive efficiency.
4. We validate the approach on both simulated jump-diffusion data and the ELEC2 electricity demand dataset [8], demonstrating consistent performance gains over single-model OCP and existing aggregation methods.

2 Preliminaries

2.1 Online Conformal Prediction Basics

We consider a data stream of covariate–response pairs $(X_t, Y_t) \in \mathcal{X} \times \mathcal{Y}$ arriving sequentially for $t = 1, 2, \ldots$. At each time t, a predictive model $\hat{f}_t : \mathcal{X} \to \mathbb{R}$ produces a residual score

$$s_t(x, y) = |y - \hat{f}_t(x)|, \tag{1}$$

Classical conformal prediction [1] constructs a prediction set for Y_t by comparing this score to a threshold q_t:

$$C_t(x) = \{y \in \mathcal{Y} : s_t(x, y) \leq q_t\}.$$

Online quantile updates.

While classical CP provides rigorous finite-sample guarantees, it fundamentally relies on the assumption of exchangeability (i.i.d. data). In practice, however, sequential

data settings often exhibit distributional shift, temporal dependence, or concept drift, conditions under which the classical guarantees may no longer hold. To address these challenges, Angelopoulos et al. [2] extended CP to the online setting, where the quantile threshold q_t is updated adaptively as new data arrive:

$$q_{t+1} = q_t + \eta_t\Big(\mathbf{1}\{Y_t \notin C_t(X_t)\} - \alpha\Big), \tag{2}$$

where $\alpha \in (0,1)$ is the target miscoverage rate and η_t is a learning rate. Intuitively, if Y_t falls outside the current prediction set, the threshold is increased to widen future intervals; otherwise, the threshold decreases, tightening the intervals.

A key refinement in this setting is the use of *decaying step sizes*,

$$\eta_t \propto t^{-1/2-\epsilon}, \quad \epsilon \in (0, 1/2), \tag{3}$$

which stabilize the change of q_t and ensure convergence of empirical coverage to the nominal level $1-\alpha$. In contrast, fixed step sizes may cause oscillations that fail to settle around the desired coverage level. This adaptive updating framework broadens the applicability of CP to realistic streaming environments where data distributions evolve over time.

Coverage.

We assess performance through the coverage probability of the conformal sets. The *instantaneous coverage* at time t is

$$\text{Coverage}_t(q_t) = \mathbf{1}\{s_t(X_t, Y_t) \le q_t\},$$

which records whether the true outcome is captured at time t. Aggregating over time gives the *long-run coverage*,

$$\widehat{\text{Coverage}}_T = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{Y_t \in C_t(X_t)\},$$

which should approach the nominal level $1-\alpha$ if the updates are well calibrated.

2.2 Conformal Online Model Aggregation (COMA)

COMA [4] aggregates K expert conformal sets online by maintaining data–adaptive weights and forming a weighted majority set each round.

Hedge and AdaHedge

Let $\ell^{(t)} \in \mathbb{R}^K_{\ge 0}$ be the per-round losses of the K experts and $L_k^{(t)} := \sum_{s=1}^{t} \ell_k^{(s)}$ their cumulative losses. The classical Hedge (exponential-weights) algorithm [6] updates the

expert weights with learning rate $\eta > 0$ as

$$w_k^{(t+1)} \;=\; \frac{\exp\big(-\eta\, L_k^{(t)}\big)}{\sum_{j=1}^{K} \exp\big(-\eta\, L_j^{(t)}\big)}, \tag{4}$$

yielding low regret relative to the best expert in hindsight.

To avoid tuning η, COMA employs the AdaHedge variant [9], which adaptively chooses the learning rate online:

$$\eta^{(t)} \;=\; \frac{\ln K}{\delta^{(1)} + \cdots + \delta^{(t-1)}}, \qquad \delta^{(t)} \;=\; h^{(t)} - m^{(t)}, \tag{5}$$

where $h^{(t)} = w^{(t)} \cdot \ell^{(t)}$ is the Hedge loss and the *mix loss* is

$$m^{(t)} \;=\; -\frac{1}{\eta^{(t)}}\, \ln\!\left(\sum_{k=1}^{K} w_k^{(t)} \exp\big\{-\eta^{(t)}\, \ell_k^{(t)}\big\}\right). \tag{6}$$

This adaptive setting makes AdaHedge parameter-free and more stable under streaming data.

Weighted majority aggregation

At round t, experts output conformal sets $\{C_1^{(t)}, \ldots, C_K^{(t)}\}$. COMA aggregates these into a single prediction set using a weighted majority rule:

$$C^{(t)} \;=\; \Big\{ y \in \mathcal{Y} \,:\, \sum_{k=1}^{K} w_k^{(t)}\, \mathbf{1}\big\{y \in C_k^{(t)}\big\} \;>\; \tfrac{1+u^{(t)}}{2} \Big\}, \qquad u^{(t)} \sim \mathrm{Unif}[0,1]. \tag{7}$$

Here $u^{(t)}$ is a random variable that determines the threshold for expert agreement. The threshold $\frac{1+u^{(t)}}{2}$ lies in $[0.5, 1]$, interpolating between a strict majority (0.5) and full consensus (1). In practice, one may either fix the threshold at some value $\rho \in [0.5, 1]$ (deterministic voting) or sample $u^{(t)} \sim \mathrm{Unif}[0,1]$ (randomized voting). The randomized rule ensures finite-sample coverage guarantees, whilethe deterministic choice allows controlled flexibility in aggregation.

Adaptivity under distribution shift

To maintain valid coverage under nonstationary conditions, COMA adopts the decaying step-size mechanism from [2], which adaptively updates the miscoverage level through Equations (2) and (3). This yields a flexible ensemble method that preserves conformal guarantees while dynamically shifting weight toward sharper experts.

3 Methodology

Classical online conformal prediction (OCP) with decaying step sizes proposed by Angelopoulos et al. [2], ensures valid coverage by updating thresholds sequentially, but

it relies on a single underlying predictor. This can lead to inefficient intervals if the chosen model is poorly calibrated or mismatched to the data. To address this limitation, we extend OCP by integrating the SL framework [3], which allows us to aggregate forecasts from multiple base learners. In this way, the conformal score (Equation (1)) is no longer tied to one model, but instead leverages a weighted ensemble that balances bias and variance across predictors. This extension targets narrower prediction intervals while maintaining the coverage guarantees of conformal prediction.

3.1 Super Learner for Model Aggregation

The Super Learner (SL) [3] is a general ensemble method that constructs a meta-learner by optimally combining a set of candidate forecasting models. It is designed to achieve performance asymptotically equivalent to the best convex combination of the base learners in terms of predictive risk. Given K learners producing point forecasts $\hat{Y}_t^{(1)}, \ldots, \hat{Y}_t^{(K)}$, the Super Learner constructs

$$\hat{Y}_t^{SL} = \sum_{k=1}^{K} \phi_k \hat{Y}_t^{(k)}, \qquad \phi \in \Delta^{K-1},$$

where $\Delta^{K-1} = \{\phi \in \mathbb{R}^K : \phi_k \geq 0,\ \sum_{k=1}^{K} \phi_k = 1\}$ is the probability simplex.

Weight estimation

The ensemble weights ϕ are obtained by minimizing cross-validated risk. Specifically, let $Z \in \mathbb{R}^{n \times K}$ denote the matrix of out-of-fold predictions from the K base learners, and let $y \in \mathbb{R}^n$ denote the observed outcomes. Then the weights are chosen as

$$\phi^\star = \arg\min_{\phi \in \Delta^{K-1}} \frac{1}{n} \sum_{t=1}^{n} \ell(y_t, Z_t \phi),$$

where ℓ is a loss function (e.g., squared error). Time-series applications require blocked or rolling cross-validation to preserve temporal dependence when generating out-of-fold predictions.

Algorithm 1 Super Learner with Cross-Validation

Input: Data $\{Y_t\}_{t=1}^n$, candidate learners $\mathcal{L} = \{L_1, \ldots, L_K\}$, number of folds V
for fold $v = 1, \ldots, V$ **do**
    Partition indices into training $\mathcal{I}_{\text{train}}^{(v)}$ and validation $\mathcal{I}_{\text{val}}^{(v)}$
    for each learner $L_k \in \mathcal{L}$ **do**
        Train L_k on $\mathcal{I}_{\text{train}}^{(v)}$
        Generate predictions on $\mathcal{I}_{\text{val}}^{(v)}$ and store in $Z_{:,k}$
    end for
end for
Solve $\phi^\star = \arg\min_{\phi \in \Delta^{K-1}} \frac{1}{n} \sum_{t=1}^n \ell(y_t, Z_t\phi)$
Retrain all $L_k \in \mathcal{L}$ on the full dataset
Compute $\hat{Y}_t^{SL} = \sum_{k=1}^K \phi_k^\star \hat{Y}_t^{(k)}$ on test data
Output: Super Learner predictions $\hat{Y}^{SL}$, weights $\phi^\star$

SL outputs an aggregated forecast $\hat{Y}_t^{SL}$ and the corresponding optimal weights $\phi^\star$. To incorporate these predictions into online conformal prediction, we compute residual scores based on Equation (1) Finally, the conformal quantile is updated sequentially using the rule in Equation (2) and Equation (3), ensuring that the resulting intervals achieve empirical coverage close to the target level $1 - \alpha$.

Integration with COMA

While COMA already provides a principled way to combine multiple base learners through adaptive re-weighting, its pool is limited to the raw experts themselves. This means COMA can only exploit individual forecasts without directly leveraging their optimal convex combination. To address this, we also consider the SL as an additional expert within the COMA framework. In this setting, COMA re-weights the SL alongside the base learners using Hedge [6] or AdaHedge, thereby balancing the strengths of individual experts with their meta-combination. This integration preserves finite-sample coverage while enabling narrower predictive intervals when the SL offers improved efficiency.

4 Experiments

4.1 Simulated Dataset

To assess the performance of our method under controlled conditions, we first generate a synthetic time series using a jump-diffusion (JD) process, a standard model in mathematical finance for capturing both continuous fluctuations and rare, abrupt shocks. The JD model evolves according to

$$S_{t+1} = S_t\Big(1 + \mu\Delta t + \sigma\Delta W_t + (e^J - 1)N_t\Big),$$

where μ is the drift, σ the volatility, ΔW_t a Gaussian increment, $N_t \sim \text{Poisson}(\lambda\Delta t)$ the jump indicator, and J the jump size distribution.

Simulation 1(Sim1) introduces *heavy-tailed shocks* by specifying $J \sim t_\nu(0, \text{scale})$, a Student's t distribution with low degrees of freedom. This allows for rare but extreme jumps, reflecting the heavy-tail behavior often observed in financial returns.

Simulation 2(Sim2) reverts to the more standard Gaussian specification with $J \sim \mathcal{N}(\mu_J, \sigma_J^2)$, producing less extreme fluctuations. For clarity, we report Sim2 results in Appendix B.

Although our simulated jump-diffusion model generates raw stock prices, we standardize the series to have zero mean and unit variance prior to model training. This is standard practice in time series forecasting to improve numerical stability, especially for neural networks, and to ensure that residual-based conformal prediction intervals are comparable across datasets and scales (see Fig. 1 for the standardized series).

Finally, to mimic measurement imperfections, we generate an observed series y_{observed} by adding heteroskedastic heavy-tailed noise to the standardized trajectory y_{true}. Specifically,

$$y_{\text{observed}} = y_{\text{true}} + \epsilon_t, \quad \epsilon_t \sim t_\nu\big(0,\, \sigma_t^2\big), \qquad \sigma_t = \exp(-2 + 0.3 z_t),\ z_t \sim \mathcal{N}(0,1).$$

Here σ_t introduces volatility clustering through a log-normal variance structure, while the Student's t distribution with low degrees of freedom induces heavy tails. All base learners are trained on y_{observed}, while evaluation is performed against the clean y_{true}.

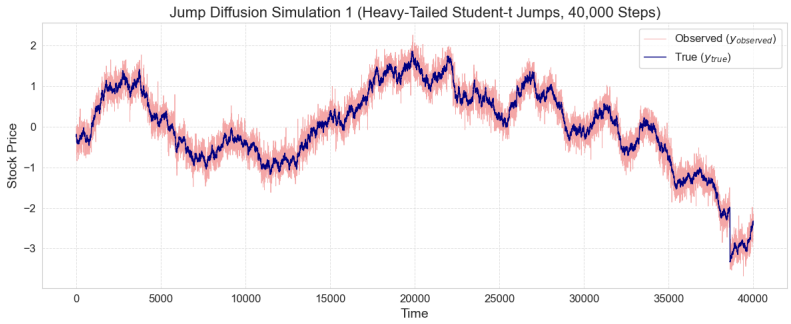

Fig. 1 Jump–diffusion model Simulation 1 with standardized stock prices.

4.1.1 Models

We evaluate our method using two complementary classes of base learners, selected to capture different levels of complexity and modeling assumptions. Within each class, we construct multiple base learners by varying their hyperparameter configurations, allowing the Super Learner to combine diverse predictors from the same model family.

Autoregressive Conditional Heteroskedasticity (ARCH) To capture volatility clustering and heavy-tailed behavior in financial time series, we employ autoregressive conditional ARCH family models. The standard GARCH formulation, proposed by Engle [10], extends classical ARMA processes by allowing the conditional variance to evolve dynamically over time. In addition to symmetric GARCH

specifications, we also include an exponential GARCH (EGARCH) variant to account for asymmetric volatility responses to positive and negative shocks. By incorporating heavy-tailed and skewed error distributions, these models can better reflect the jump–diffusion characteristics of the data. The specific configurations used are summarized in Table 1.

Model	Configuration
GARCH(1,1)	AR(1) mean, Student-t errors, re-estimated every 50 steps
EGARCH(1,1)	AR(1) mean, Student-t errors, re-estimated every 20 steps
GARCH(2,2)	AR(2) mean, skew-t errors, re-estimated every 50 steps

Table 1 ARCH-family base learner configurations used in the SL ensemble on Simulation 1.

Long Short-Term Memory (LSTM) To capture nonlinear dependencies and long-range temporal patterns, we include three LSTM architectures with distinct depths, activation functions, and regularization. (Long short-term memory (LSTM) architecture originally proposed by Hochreiter and Schmidhuber [11]. The first is a deep tanh network with two stacked layers (256 and 64 units), the second is a shallower ReLU network with stronger regularization, and the third is a compact single-layer ReLU model optimized for speed. All models were trained with a sequence length of 15 using the Adam optimizer with dropout to mitigate overfitting, (see details in Table 2).

Model	Layers	Activation	Dropout	Learning Rate	Epochs	Batch Size
LSTM 1	$256 \to 64$	tanh	0.1	0.0005	6	16
LSTM 2	$128 \to 32$	ReLU	0.1	0.0002	10	16
LSTM 3	32	ReLU	0.1	0.001	10	32

Table 2 LSTM-family base learner configurations used in the SL ensemble on Simulation 1.

4.1.2 Coverage Guarantee

A fundamental prerequisite for our conformal prediction methods is that they achieve the target coverage level of $1-\alpha$ where $\alpha = 0.1$. For clarity and conciseness, the main text reports results only from ARCH-family models, while results for LSTM-based models are presented in the Appendix A.

Table 3 shows that all ARCH-family base learners, the SL ensemble, COMA with three base learners, and COMA augmented with the SL maintain empirical coverage close to the nominal 0.9. This ensures that subsequent comparisons are made between valid conformal predictors.

Method	Mean Coverage
ARCH(1,1)	0.9091
EGARCH(1,1)	0.9016
GARCH(2,2)	0.8966
Super Learner	0.8995
COMA (plain)	0.8947
COMA with SL	0.8990

Table 3 Mean empirical coverage on Simulation 1 (ARCH-family models, AdaHedge).

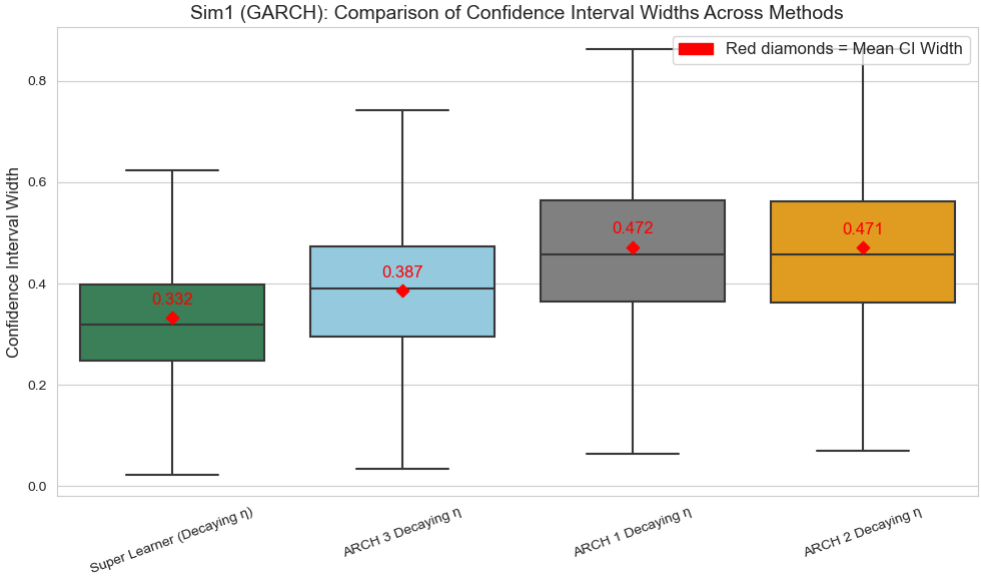

Fig. 2 Confidence interval widths of ARCH-family base learners vs. SL on Simulation 1.

4.1.3 COMA with AdaHedge and Hedge

AdaHedge Algorithm

Having established valid coverage guarantees, we next evaluate the *width* of the prediction intervals. On average, the base learners yield relatively wide intervals (GARCH(1,1): 0.472, EGARCH(1,1): 0.471, GARCH(2,2): 0.387), whereas the SL attains a substantially lower mean width of 0.3320. As shown in Figure 2, the SL also exhibits a more concentrated distribution with a lower median, indicating improved stability relative to individual models. Together, these results demonstrate that ensembling heterogeneous predictors enhances efficiency without compromising coverage.

To further evaluate performance, we compare the SL with COMA implemented under AdaHedge, which aggregates the same set of base learners in an online fashion.

Table 4 Mean confidence interval (CI) widths for COMA and SL on Simulation 1 using ARCH-family models with AdaHedge.

Method	COMA (Deterministic)	COMA (Randomized)	Super Learner (Decaying)
Mean CI Width	0.3865	0.3865	0.3322

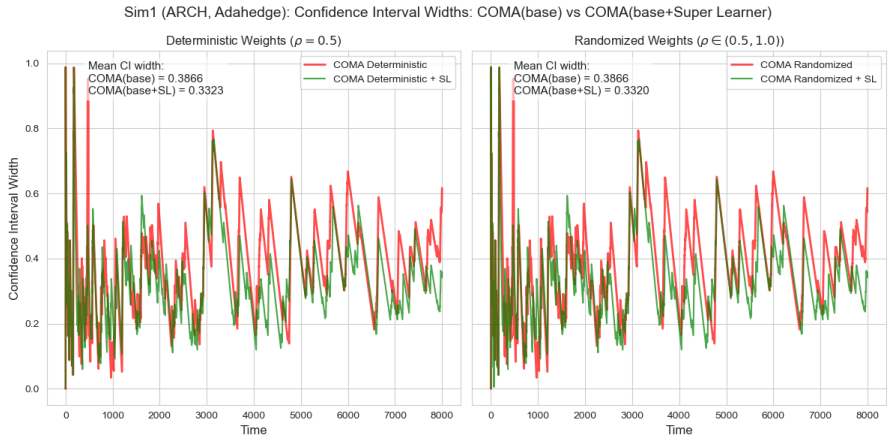

Fig. 3 Distribution of confidence interval (CI) widths on Simulation 1 using ARCH-based models with the AdaHedge.

As shown in Table 4, both deterministic and randomized COMA variants yield mean CI widths of approximately 0.3865, while the SL achieves a narrower mean width of 0.3322. This indicates that the SL provides more efficient intervals than COMA while maintaining comparable coverage guarantees.

We then extend COMA by including the SL itself as an additional expert alongside the ARCH-based models. As described in Section 2.2, the aggregation depends on the consensus threshold ρ, which controls how much agreement among experts is required. Figure 3 reports CI widths under deterministic ($\rho = 0.5$) and randomized ($\rho \in (0.5, 1.0)$) weighting. In this heavy-tailed setting, COMA+SL consistently yields narrower intervals than COMA(base), with improvements most pronounced in the tails where extreme jumps occur. On average, CI width decreases from 0.3865 to 0.3323 under deterministic weights (a 14.0% reduction) and from 0.3865 to 0.3320 under randomized weights (a 14.1% reduction). These results demonstrate that adding the Super Learner stabilizes aggregation and improves efficiency in volatile regimes.

For completeness, we repeat the same analysis with Hedge in Appendix C.1, where we observe identical trends.

4.2 Real-World Data: Electricity Demand (ELEC2)

To evaluate performance in a real-world setting, we replicate the electricity demand experiment from Gasparin and Ramdas [4], using the `ELEC2` dataset [8]. This dataset

records electricity pricing and demand in New South Wales and Victoria, with transfer quantities representing the electricity exchanged between the two states. Following the setup in [4], we focus on a time window between 9:00 am and 12:00 pm and construct three predictive models: an autoregressive AR(2) model, an AR(2) model with an additional covariate, and a covariate-only regression. As shown in Table 5,

Method	**Mean Coverage**
AR(2)	0.9005
AR(2) + Covariate	0.9005
Only Covariate	0.9008
Super Learner (Decaying)	0.9002
COMA (Plain)	0.8973
COMA with SL	0.8991

Table 5 Mean empirical coverage on `ELEC2` dataset.

all methods satisfy the nominal coverage guarantee of $1 - \alpha = 0.9$.

Table 6 Mean confidence interval (CI) widths for COMA and SL on `ELEC2` using AdaHedge.

Method	COMA (Deterministic)	COMA (Randomized)	Super Learner (Decaying)
Mean CI Width	0.3463	0.3456	0.3326

With coverage guarantee guarantees established, we compare SL with COMA implemented Adahedge. Table 6 reports the mean confidence interval widths: about 0.3463 for COMA (Deterministic), 0.3456 for COMA (Randomized), and 0.3326 for the Super Learner. These results indicate that the Super Learner produces slightly narrower intervals than COMA, while preserving comparable coverage guarantees.

We then aggregate the base learners using COMA with both deterministic and randomized weighting, and compare against our extension that augments COMA with the Super Learner (COMA+SL). Figure 4 reports the distribution of CI widths on the `ELEC2` dataset. COMA+SL consistently yields narrower intervals than COMA alone while preserving valid coverage. Under deterministic weighting ($\rho = 0.5$), the mean width decreases from 0.3463 to 0.3327 (a 3.9% reduction), with similar gains under randomized weighting. Appendix D provides results with smoothed confidence interval comparison.

These findings confirm that the benefits of integrating the SL extend to real-world electricity demand forecasting, reinforcing its role as a strong expert that COMA can effectively exploit to achieve efficiency gains. For completeness, we also repeat the same analysis with Hedge in Appendix C.2, where we observe identical trends.

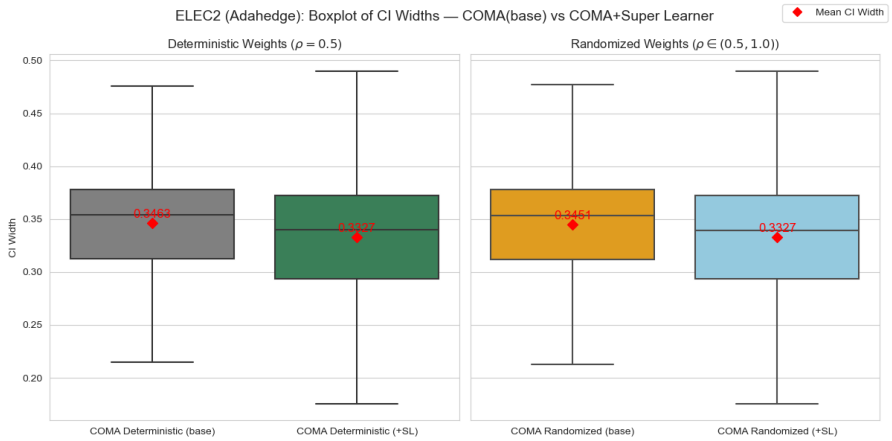

Fig. 4 Distribution of confidence interval (CI) widths on the `ELEC2` dataset.

5 Discussion

This paper examined online conformal prediction and model aggregation for time series forecasting, focusing on COMA and the SL. Our results highlight three main insights. First, the SL consistently achieves narrower confidence intervals than both individual base learners and COMA, with modest but robust efficiency gains across simulated and real datasets. Second, when incorporated as an expert within COMA, the SL dominates the aggregation, yielding intervals that closely track its own, showing COMA's ability to adapt to strong experts. Third, we demonstrate that useful diversity can be created even within a single model family by varying hyperparameters and specifications. For example, ARCH-type models with different volatility structures and error distributions, and LSTM architectures with distinct depths, activations, and dropout levels, all provided complementary signals for aggregation.

Our study has several limitations. The base learners were restricted to a narrow set of families, with diversity induced through hyperparameter variation rather than fundamentally different model classes. We also evaluated efficiency primarily via mean confidence interval width, leaving aside conditional coverage, robustness to distributional shifts, and computational trade-offs. Finally, experiments were limited to jump–diffusion simulations and the `ELEC2` dataset, so broader validation is needed.

Future work should extend ensemble design beyond family-based hyperparameter variation to include structurally different learners (e.g., tree-based or attention models), explore richer efficiency metrics, and develop theory to characterize COMA when strong ensemble predictors such as the SL are included.

In sum, we show that meta-learning and online aggregation can deliver efficiency gains in conformal prediction, even when base learners come from the same family. This underscores the potential of combining heterogeneous predictors and motivates further exploration of broader model classes and theoretical guarantees.

References

[1] Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York, NY (2005). https://doi.org/10.1007/b106715

[2] Angelopoulos, A.N., Barber, R.F., Bates, S.: Online conformal prediction with decaying step sizes. arXiv preprint arXiv:2402.01139 (2024) https://doi.org/10.48550/arXiv.2402.01139

[3] Laan, M.J., Polley, E.C., Hubbard, A.E.: Super learner. Statistical Applications in Genetics and Molecular Biology **6**(1), 25 (2007) https://doi.org/10.2202/1544-6115.1309

[4] Gasparin, M., Ramdas, A.: Conformal online model aggregation (2024). https://arxiv.org/abs/2403.15527

[5] Gibbs, I., Candès, E.J.: Conformal prediction in the adversarial setting. arXiv preprint arXiv:2106.00170 (2021)

[6] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1), 119–139 (1997) https://doi.org/10.1006/jcss.1997.1504

[7] Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press, Cambridge, UK (2006). https://doi.org/10.1017/CBO9780511546921

[8] Harries, M.: Splice-2 comparative evaluation: Electricity pricing. Technical report, University of New South Wales (2003)

[9] Rooij, S., Erven, T., Grünwald, P., Koolen, W.M.: Follow the leader if you can, hedge if you must. Journal of Machine Learning Research **15**(1), 1281–1316 (2014)

[10] Engle, R.: Garch 101: The use of arch/garch models in applied econometrics. Journal of Economic Perspectives **15**(4), 157–168 (2001) https://doi.org/10.1257/jep.15.4.157

[11] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) https://doi.org/10.1162/neco.1997.9.8.1735

A LSTM results for Simulation 1

To complement the ARCH-family analysis in the main text, we also evaluate Simulation 1 using LSTM-family base learners. Table 7 confirms that all methods maintain valid coverage guarantees close to the nominal level of 0.9. As shown in Table 8,

the SL achieves narrower confidence intervals (mean width 0.2907) than both COMA variants (0.3347 deterministic, 0.3343 randomized), demonstrating its efficiency gains while preserving coverage.

In addition, Figure 5 shows the distribution of CI widths. Augmenting COMA with the SL yields further reductions in interval width relative to COMA alone, under both deterministic and randomized weighting schemes. This result parallels the findings from the ARCH-family models and reinforces the consistency of our conclusions across different base learner classes in the heavy-tailed jump–diffusion setting.

Table 7 Mean empirical coverage on Simulation 1 (LSTM-family models, AdaHedge).

Method	**Mean Coverage**
LSTM 1	0.8936
LSTM 2	0.8907
LSTM 3	0.8887
Super Learner	0.8961
COMA (Deterministic)	0.8901
COMA with SL (Deterministic)	0.8941

Table 8 Mean CI widths for COMA and Super Learner on Simulation 1 using LSTM-family models with AdaHedge.

Method	COMA (Deterministic)	COMA (Randomized)	Super Learner (Decaying)
Mean CI Width	0.3347	0.3343	0.2907

B Robustness to Simulation Parameters

In the main text (Section 4.1), we evaluated the SL and COMA under a jump–diffusion process with heavy-tailed Student-t jumps, representing a market environment with extreme shocks and volatility clustering.

To assess robustness, we repeated the experiment under an alternative specification with Gaussian jump sizes: $\mu = 0.05$, $\sigma = 0.2$, $\lambda = 0.1$, $\mu_J = -0.1$, and $\sigma_J = 0.3$. This setting corresponds to a more conventional market regime with moderate fluctuations and less frequent extreme events. Following the same empirical protocol as in Simulation 1, we composed the base learners using the same model family (LSTM) but with diverse hyperparameter configurations to capture different aspects of volatility and jump dynamics in the Gaussian jump–diffusion setting. For clarity and conciseness, we omit detailed descriptions here; the full configurations are summarized in Table 9.

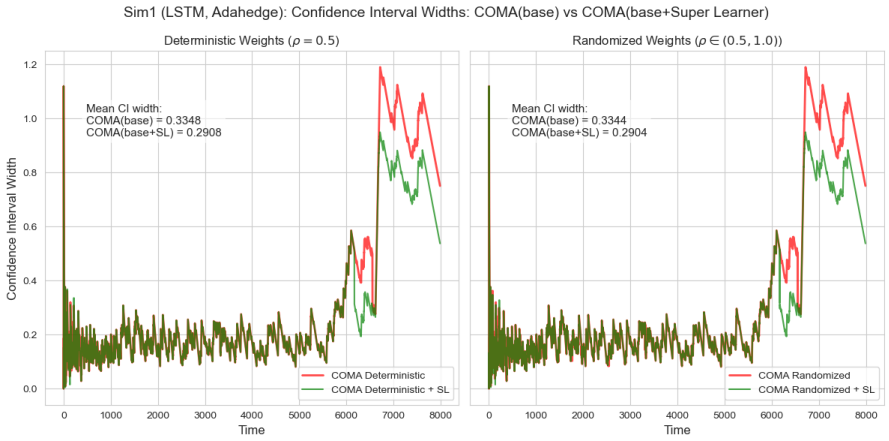

Fig. 5 Distribution of CI widths in Simulation 1 using LSTM-based models with AdaHedge.

Model	Layers	Activation	Dropout	Learning Rate	Epochs	Batch Size
LSTM 1	$256 \to 128$	tanh	0.25	0.0002	10	32
LSTM 2	$128 \to 64$	ReLU	0.20	0.0007	12	16
LSTM 3	32	ReLU	0.10	0.002	15	64

Table 9 LSTM-family base learner configurations used in Simulation 2 (Gaussian jump–diffusion).

Tables 10 and 11 summarize the results. All methods achieve empirical coverage close to the nominal level $1 - \alpha = 0.9$, confirming that the validity guarantees of conformal prediction hold in this Gaussian setting. Meanwhile, the SL produces slightly narrower mean intervals (0.1704) than COMA (Deterministic 0.1727, Randomized 0.1725).

We also extend COMA by including the Super Learner as an additional expert. Figure 6 reports CI widths under both deterministic ($\rho = 0.5$) and randomized ($\rho \in (0.5, 1.0)$) weights. While the shrinkage effect is smaller than in the heavy-tailed case, COMA+SL consistently achieves modest reductions in CI width. This indicates that the efficiency gains observed in Simulation 1 persist in more stable market regimes, though with reduced magnitude.

For completeness, we repeat the same analysis with Hedge in Appendix C.1, where we observe identical trends.

C COMA with Hedge

In this appendix, we report additional results obtained using the classical Hedge algorithm in place of AdaHedge. The analysis covers both the simulation studies and the `ELEC2` dataset. Following Freund and Schapire [6], Cesa-Bianchi and Lugosi [7],the Hedge algorithm was implemented with a fixed learning rate of $\eta = 0.1$, as defined in Equation 4, consistent with the setup in Gasparin and Ramdas [4].

Table 10 Mean empirical coverage on Simulation 2 with Gaussian jumps.

Method	**Mean Coverage**
LSTM 1	0.9003
LSTM 2	0.8984
LSTM 3	0.8990
Super Learner	0.8999
COMA (Deterministic)	0.8998
COMA with SL (Deterministic)	0.9006

Table 11 Mean CI widths for COMA and Super Learner on Simulation 2 (Gaussian JD).

Method	COMA (Deterministic)	COMA (Randomized)	Super Learner (Decaying)
Mean CI Width	0.2622	0.2620	0.2373

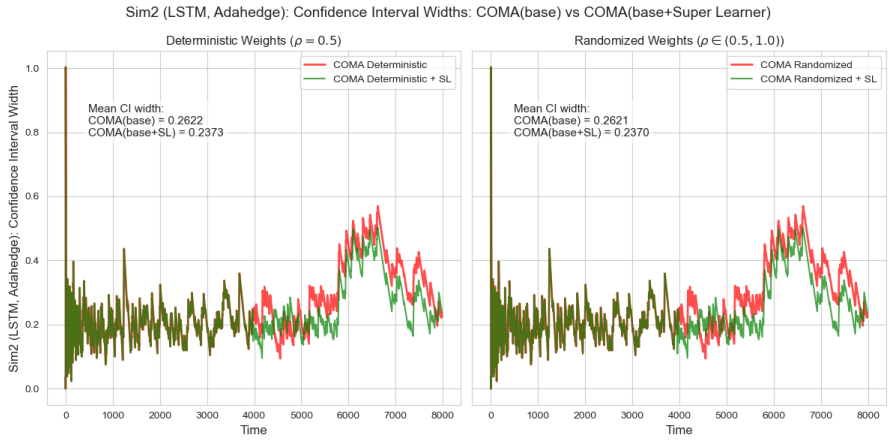

Fig. 6 CI widths in Simulation 2 (Gaussian JD) using LSTM-based models with AdaHedge.

C.1 Simulation Study

In the main analysis we used AdaHedge as the aggregation rule within COMA. Figures 7–9 summarize the confidence interval (CI) widths across both simulation settings and model families.

For **Simulation 1**, consistent patterns emerge across the two model classes. With ARCH-based learners (Figure 7), incorporating the SL reduces mean CI width from 0.3874 to 0.3320 under deterministic weights ($\rho = 0.5$), and from 0.3835 to 0.3276 under randomized weights ($\rho \in (0.5, 1.0)$). Analogous improvements are observed for LSTM-based learners (Figure 8), where mean widths decrease from 0.3349 to 0.2912 (deterministic) and from 0.3251 to 0.2809 (randomized).

For **Simulation 2** with LSTM-based learners (Figure 9), the SL again shrinks intervals, from 0.2629 to 0.2376 under deterministic aggregation and from 0.2590 to 0.2275 under randomized aggregation.

Overall, across both simulations and model families, augmenting COMA with the SL consistently narrows prediction intervals without sacrificing coverage.

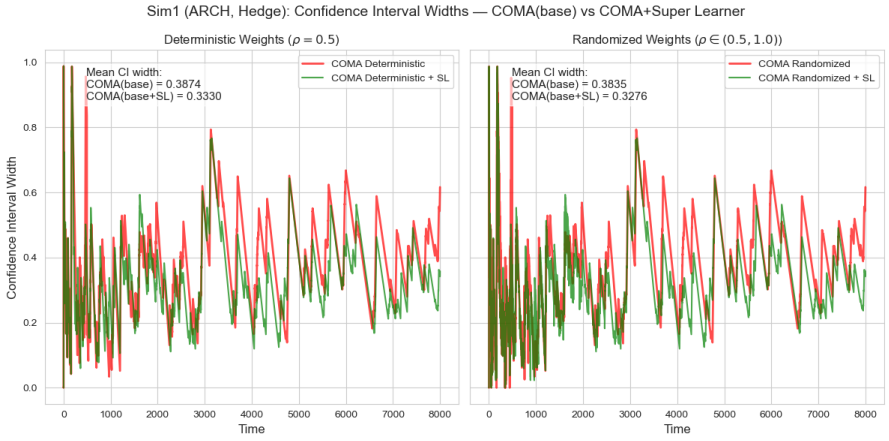

Fig. 7 Confidence interval (CI) widths on Simulation 1 using ARCH-based models with the Hedge algorithm.

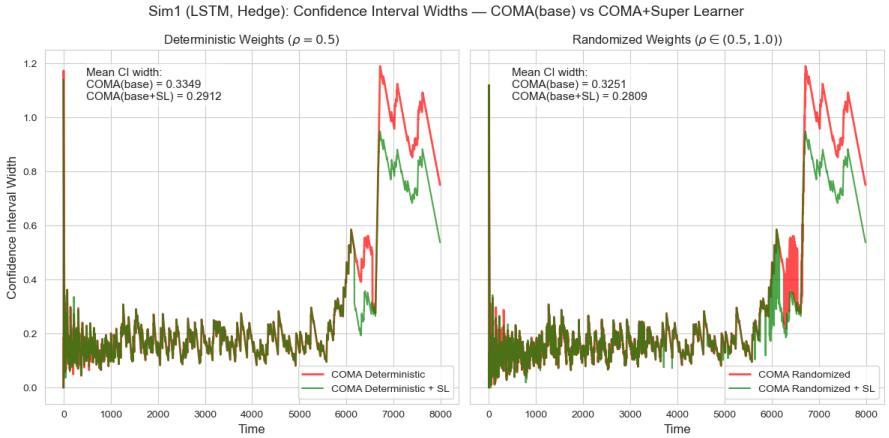

Fig. 8 Confidence interval (CI) widths on Simulation 1 using LSTM-based models with the Hedge algorithm.

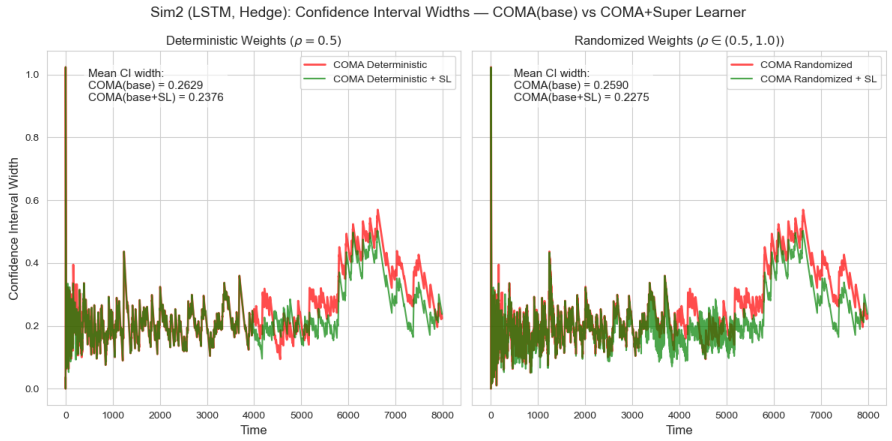

Fig. 9 Confidence interval (CI) widths on Simulation 2 using LSTM-based models with the Hedge algorithm.

C.2 ELEC2 Dataset

As with the simulation study, we applied the Hedge algorithm to the real-world `ELEC2` dataset, comparing COMA with base learners against COMA augmented with the Super Learner (COMA+SL) under both deterministic and randomized weighting. Figure 10 shows that adding the SL yields consistently narrower confidence intervals. In the deterministic case ($\rho = 0.6$), the mean CI width decreases from 0.3431 to 0.3326 (a reduction of 0.0105). In the randomized case ($\rho \in (0.5, 1.0)$), the width decreases from 0.3309 to 0.3291 (a reduction of 0.0038). While the gains are modest, they confirm that the efficiency benefit of including the SL extends beyond simulated settings to real-world electricity demand forecasting.

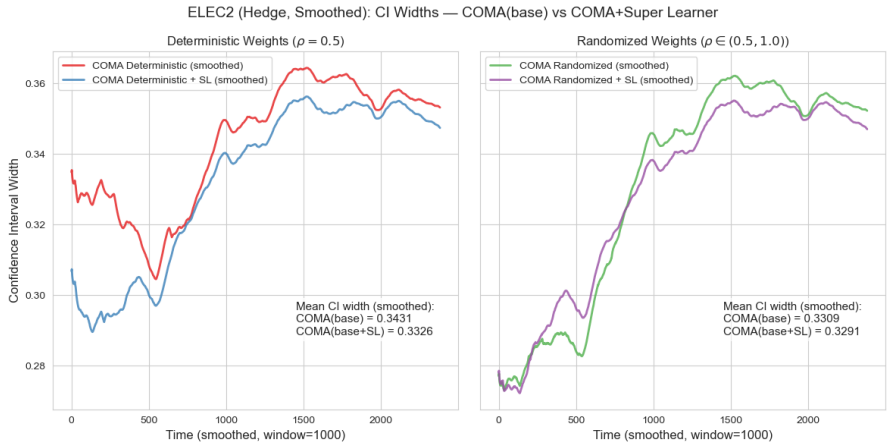

Fig. 10 Smoothed confidence interval (CI) widths on the `ELEC2` dataset (window size = 1000) using Hedge.

D More results of ELEC2 Dataset

As in both the Adahedge and Hedge experiments, we generated the smooth confidence interval which can better show comparison.

Figure 11 reports the smoothed CI widths for the `ELEC2` dataset under both deterministic weighting ($\rho = 0.5$) and randomized weighting ($\rho \in (0.5, 1.0)$). The benefit of incorporating the Super Learner (SL) into COMA is clearly visible: across both weighting schemes, the COMA+SL curves lie consistently below those of COMA with base learners alone, indicating narrower prediction intervals on average.

These results reinforce the findings in the main analysis and in Appendix C, confirming that statistical stacking through the SL yields efficiency gains not only on simulated data but also in real-world electricity demand forecasting.

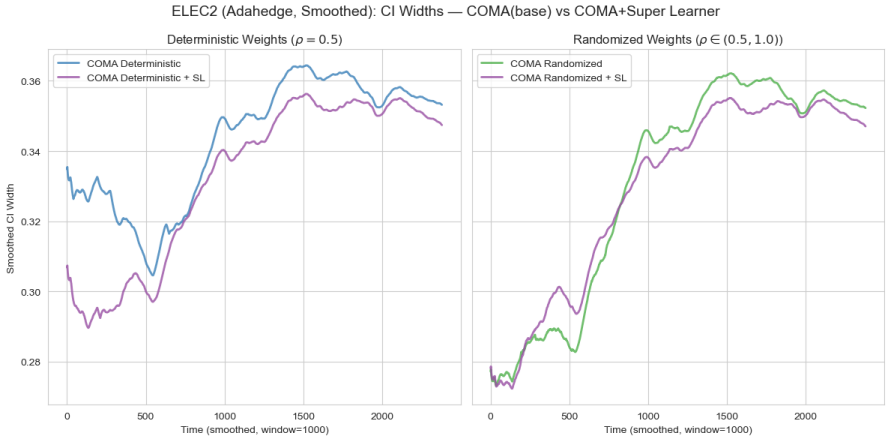

Fig. 11 Smoothed confidence interval (CI) widths on the `ELEC2` dataset (window size = 1000) using Adahedge algorithm