The Meta-Learning Gap: Combining Hydra and Quant for Large-Scale Time Series Classification

Urav Maniar

Faculty of Information Technology, Monash University, Clayton, 3168, Victoria, Australia.

Corresponding author(s). E-mail(s): urav06@gmail.com;

Abstract

Time series classification faces a fundamental trade-off between accuracy and computational efficiency. While comprehensive ensembles like HIVE-COTE 2.0 achieve state-of-the-art accuracy, their 340-hour training time on the UCR benchmark renders them impractical for large-scale datasets. We investigate whether targeted combinations of two efficient algorithms from complementary paradigms can capture ensemble benefits while maintaining computational feasibility. Combining Hydra (competing convolutional kernels) and Quant (hierarchical interval quantiles) across six ensemble configurations, we evaluate performance on 10 large-scale MONSTER datasets (7,898 to 1,168,774 training instances). Our strongest configuration improves mean accuracy from 0.829 to 0.836, succeeding on 7 of 10 datasets. However, prediction-combination ensembles capture only 11% of theoretical oracle potential, revealing a substantial meta-learning optimization gap. Feature-concatenation approaches exceeded oracle bounds by learning novel decision boundaries, while prediction-level complementarity shows moderate correlation with ensemble gains. The central finding: the challenge has shifted from ensuring algorithms are different to learning how to combine them effectively. Current meta-learning strategies struggle to exploit the complementarity that oracle analysis confirms exists. Improved combination strategies could potentially double or triple ensemble gains across diverse time series classification applications.

Keywords: time series classification, ensemble methods, stacked generalization, meta-learning, complementarity analysis, oracle ensemble, CAWPE, Hydra, Quant, large-scale datasets

1

1 Introduction

Time series classification confronts a fundamental tension between accuracy and computational efficiency. Modern applications generate time series data at unprecedented scales: wearable sensors producing millions of activity sequences, satellite systems capturing continuous Earth observations, and industrial monitoring generating real-time sensor streams. Classification algorithms must process this data with both high accuracy and practical efficiency, yet current approaches struggle to achieve both simultaneously.

State of the art accuracy in time series classification comes from comprehensive ensemble methods like HIVE-COTE 2.0, which systematically combines algorithms from multiple representational paradigms (Middlehurst et al. 2021). This multi-paradigm approach captures the principle that different problem types favor different algorithmic strategies (Bagnall et al. 2017). However, achieving this accuracy requires prohibitive computational cost: HIVE-COTE 2.0 requires 340 hours to train on the 112-dataset UCR benchmark compared to 2.85 hours for efficient single-algorithm approaches like ROCKET (Middlehurst et al. 2021). This 120-fold computational overhead renders comprehensive ensembles impractical for large-scale real-world datasets, where training times would extend to weeks or months.

Recent algorithmic advances demonstrate that efficiency and accuracy need not be entirely opposed. Hydra (Dempster et al. 2023), using competing convolutional kernels to identify dominant local patterns, and Quant (Dempster et al. 2024), extracting quantile features from hierarchical intervals, both achieve competitive accuracy while training in under one hour on benchmark datasets. These algorithms represent distinct paradigms: Hydra captures pattern frequencies through kernel competition while Quant summarizes distributional statistics, suggesting potential complementarity. However, almost all prior work on ensemble methods for time series classification has focused on the UCR archive, a collection of relatively small benchmark datasets. Whether ensemble strategies remain effective on larger, real-world datasets from repositories like MONSTER (Dempster et al. 2025a) remains essentially unknown.

This work investigates a fundamental question: can targeted combination of two efficient algorithms from complementary paradigms capture the accuracy benefits of comprehensive ensembles while maintaining computational feasibility for large-scale problems? We systematically explore ensemble strategies for combining Hydra and Quant across six distinct configurations, ranging from simple feature concatenation to sophisticated stacked generalization with out-of-fold prediction generation. Rather than assuming complementarity, we empirically validate it through comprehensive analysis measuring both feature-level correlation and prediction-level error independence.

Our investigation proceeds in three stages. First, we quantify complementarity between Hydra and Quant across 10 large-scale MONSTER datasets, measuring feature space correlation, error pattern independence, and oracle ensemble potential (the theoretical best-case accuracy when at least one algorithm is correct, providing an upper bound for prediction-combination strategies). Second, we systematically evaluate six ensemble configurations representing the major combination strategies: feature concatenation with linear and non-linear classifiers, asymmetric stacking that

2

augments Quant features with Hydra predictions, symmetric stacking that combines out-of-fold predictions from both algorithms, and weighted averaging through a Simplified Cross-validation Accuracy Weighted Probabilistic Ensemble (Simplified CAWPE). Third, we analyze what determines ensemble success, correlating complementarity metrics with actual performance gains.

The central research question: can we close the gap between theoretical oracle potential and actual ensemble performance? If so, which meta-learning strategies effectively exploit complementarity between efficient algorithms?

2 Related Work

Time series classification confronts a fundamental challenge: different problem types favor different algorithmic approaches (Bagnall et al. 2017). While combining diverse algorithms achieves state-of-the-art accuracy, computational costs often exceed 300 hours (Middlehurst et al. 2021). Recent fast algorithms reduce this to under 2 hours but remain single-paradigm (Dempster et al. 2020, 2023, 2024). Moreover, almost all ensemble work focuses on the UCR archive; effectiveness on larger datasets remains unknown (Middlehurst et al. 2024).

2.1 Paradigm Diversity in Time Series Classification

The comprehensive evaluation by Bagnall et al. (2017) established that time series classification problems exhibit substantial heterogeneity, identifying five primary algorithmic paradigms (Bagnall et al. 2017, Section 2, pp. 610–611): whole series methods using elastic distance measures like dynamic time warping (Bagnall et al. 2017, p. 611), interval based methods extracting statistical features from subseries, shapelet based approaches discovering phase independent discriminative subsequences, dictionary methods classifying based on recurring symbolic patterns, and combination approaches integrating multiple paradigms.

The critical empirical finding is that no single paradigm dominates across all problem types (Bagnall et al. 2017, Table 11, p. 648). Performance differences between optimal and suboptimal paradigms range from 5% to 15%, with gaps exceeding 14% when dictionary methods are optimal compared to whole series approaches (Bagnall et al. 2017, Table 11, p. 648). This paradigm diversity establishes the theoretical motivation for ensemble methods: combining approaches from different paradigms can capture discriminatory features that no single method detects.

2.2 Ensemble Approaches

2.2.1 HIVE-COTE: Comprehensive Multi-Paradigm Combination

HIVE-COTE 2.0 achieves state of the art performance by combining four component classifiers from distinct representational domains (Middlehurst et al. 2021, p. 3): Shapelet Transform Classifier (phase independent patterns), Temporal Dictionary Ensemble (frequency features), Diverse Representation Canonical Interval Forest (interval statistics), and Arsenal (convolutional features). Components are combined via CAWPE weighting (Large et al. 2019), which raises estimated accuracy to

power $\alpha = 4$ to magnify performance differences. HIVE-COTE 2.0 ranks first on 112 UCR datasets but requires 340 hours training compared to 2.85 hours for ROCKET, a 119-fold increase. This computational expense motivates development of efficient algorithms that capture complementarity without prohibitive cost.

2.2.2 Theoretical Foundations

Ensemble theory establishes that combining diverse algorithms can exceed individual performance when base learners capture complementary discriminatory features. Stacked generalization (Wolpert 1992) provides the meta-learning framework: ensembles work by deducing the biases of base algorithms with respect to the learning set (Wolpert 1992, p. 241), training a meta-learner to identify when each algorithm produces reliable predictions. The framework partitions training data to generate out of fold predictions, preventing meta-learners from memorizing training errors (Wolpert 1992, p. 244).

Effective combination requires base algorithms that span the space of generalizers and are mutually orthogonal (Wolpert 1992, p. 256), capturing different problem aspects and making independent errors. The paradigm diversity demonstrated by Bagnall et al. (2017, Table 11, p. 648) provides empirical support: different algorithmic families excel on different problem types, creating opportunity for complementary combination. CAWPE's exponential weighting exploits this by allowing stronger performers on specific problems to dominate while preserving diversity (Large et al. 2019, p. 1675), balancing trust in accuracy estimates against hedging through combination.

2.3 Fast Algorithms for Time Series Classification

2.3.1 ROCKET: Random Convolutional Kernels

ROCKET (Dempster et al. 2020) demonstrated that random convolutional kernels combined with simple linear classifiers achieve state of the art accuracy at a fraction of traditional computational expense. The method generates 10,000 random kernels with varying parameters and applies two pooling operators: proportion of positive values (PPV), capturing the fraction of the series matching kernel patterns, and global max pooling (Dempster et al. 2020, Section 3, pp. 1462–1463). While individual random kernels provide approximate feature detection, the ensemble of diverse kernels collectively captures discriminatory patterns (Dempster et al. 2020, p. 1455). ROCKET trained in under 2 hours for 85 datasets compared to over 11 days for HIVE-COTE (Dempster et al. 2020, p. 1455), establishing random convolution as a viable paradigm.

2.3.2 Hydra: Competing Kernels

Building on ROCKET's foundation, Hydra introduces a competing kernel mechanism bridging convolutional and dictionary approaches (Dempster et al. 2023). The method organizes kernels into $g = 64$ groups of $k = 8$ kernels each (default configuration from the original implementation) (Dempster et al. 2023, Section 3.2, p. 1788). At each time point, only the kernel with largest magnitude response within each group contributes to the feature representation. The algorithm counts how often each kernel wins through

hard counting (binary indicator) or soft counting (accumulated magnitude) (Dempster et al. 2023, Section 3.2, pp. 1789–1790).

This per time point competition connects to dictionary methods: just as dictionary approaches count word frequency, Hydra counts dominant kernel pattern frequency (Dempster et al. 2023, p. 1786). By varying group size g and kernels per group k, the method interpolates between ROCKET's global pooling and dictionary style competition (Dempster et al. 2023, Figure 2, p. 1791). Hydra preserves temporal structure that global pooling discards, achieving accuracy comparable to dictionary methods while training in 31 minutes on 106 datasets (Dempster et al. 2023, p. 1781). Combining Hydra with MultiROCKET produces ensembles competitive with HIVE-COTE (Dempster et al. 2023, pp. 1796–1797).

2.3.3 Quant: Interval Quantiles

Quant represents a minimalist interval method using quantile features from fixed dyadic intervals (Dempster et al. 2024). The method recursively divides the series at depth $d = 6$ (default configuration), creating hierarchical decompositions at scales $\{2^0, 2^1, \ldots, 2^5\}$ (Dempster et al. 2024, p. 2384). For each interval of length m, Quant computes $k = 1 + \lfloor (m-1)/4 \rfloor$ evenly spaced quantiles, subtracting the interval mean from every second quantile to encode both absolute distributions and shift invariant shape at no additional cost (Dempster et al. 2024, p. 2385, Figure 4, p. 2386). Features are extracted from four representations: original series, first order differences, second order differences, and Fourier transform magnitude (Dempster et al. 2024, Section 3.1, p. 2384).

Quantiles subsume traditional summary statistics (Dempster et al. 2024, p. 2378) while enabling the classifier to handle interval selection through 10% feature sampling at each split (Dempster et al. 2024, Section 4.2.6, p. 2387). Quant trains in under 15 minutes on 142 datasets (Dempster et al. 2024, p. 2390), ranking sixth in the 2024 comprehensive comparison (Middlehurst et al. 2024, Table 14, p. 2008).

2.4 Strategies for Combining Algorithms

Three primary approaches exist for combining base algorithms into ensembles.

Feature concatenation merges feature matrices $\mathbf{H} \in \mathbb{R}^{n \times d_H}$ and $\mathbf{Q} \in \mathbb{R}^{n \times d_Q}$ into $[\mathbf{H} \mid \mathbf{Q}]$ for a single classifier. This allows learning interactions between heterogeneous features but creates high dimensional spaces requiring classifiers with appropriate feature sampling strategies.

Stacked generalization (Wolpert 1992) operates on algorithm predictions rather than raw features. Training must use out of fold predictions to prevent overfitting (Wolpert 1992, p. 244): partition data into K folds, train base algorithms on $K - 1$ folds, generate predictions for the held out fold, then train the meta-learner on these out of fold predictions. This ensures the meta-learner learns generalizable combination patterns rather than memorizing training errors. The framework accommodates asymmetric designs where deterministic transformations (Quant features) combine with fitted model predictions (Hydra logits).

Weighted averaging through CAWPE requires no meta-learner training (Large et al. 2019). Each algorithm's cross-validation accuracy determines weight w_j, raised to power $\alpha = 4$ to amplify competence differences (Large et al. 2019, p. 1675):

$$P(y = i \mid \mathbf{x}) = \frac{\sum_j w_j^\alpha \cdot P_j(y = i \mid \mathbf{x})}{\sum_j w_j^\alpha} \tag{1}$$

This balances trusting accuracy estimates against hedging through combination, performing well on small training sets (Large et al. 2019, Table 3, p. 1693).

2.5 Research Gap

The preceding sections establish that paradigm diversity motivates ensembles (Section 2.1), comprehensive ensembles achieve state of the art accuracy but require 340 hours on the UCR benchmark (Section 2.2), efficient algorithms like Hydra and Quant train in under 1 hour (Section 2.3), and theoretical frameworks exist for combination (Section 2.4). What remains unexplored is whether targeted combination of two efficient algorithms from complementary paradigms captures ensemble benefits while maintaining efficiency.

Hydra (competing convolutional kernels identifying dominant local patterns) and Quant (quantile based distributional statistics from hierarchical intervals) represent distinct paradigms mirroring HIVE-COTE's multi-paradigm coverage through two efficient components. Evidence that combining efficient methods shows promise exists: MultiROCKET combined with Hydra achieves 0.884 accuracy, ranking second and approaching HIVE-COTE (Middlehurst et al. 2024, Table 14, p. 2008). However, that combination employs two convolution methods from the same paradigm. Whether cross-paradigm combination of Hydra and Quant yields similar benefits, and which combination strategy (concatenation, stacking, or CAWPE) proves most effective, remains an empirical question this work addresses.

3 Methodology

We develop systematic ensembles combining Hydra and Quant through comprehensive exploration of the ensemble design space. Prior work established that TSC problems benefit from combining classifiers built on different representations (Middlehurst et al. 2021, p. 3213), as different paradigms capture discriminatory features in distinct data domains. We validate this principle for Hydra and Quant through complementarity analysis before constructing ensembles, providing empirical foundation for the combination strategies.

3.1 Complementarity Analysis Framework

To validate that Hydra and Quant capture orthogonal discriminatory information, we measure complementarity at both feature level (representation similarity) and prediction level (error independence). These metrics quantify the diversity that ensemble theory identifies as necessary for performance gains (Wolpert 1992).

3.1.1 Feature-level complementarity

Let $\mathbf{H} \in \mathbb{R}^{n \times d_H}$ and $\mathbf{Q} \in \mathbb{R}^{n \times d_Q}$ represent Hydra and Quant feature matrices extracted from n test samples. We compute the cross-correlation matrix between all feature pairs by concatenating features $[\mathbf{H} \mid \mathbf{Q}]$ and computing $\mathbf{C} = \mathrm{corr}([\mathbf{H} \mid \mathbf{Q}]^T)$, where $\mathbf{C} \in \mathbb{R}^{(d_H+d_Q) \times (d_H+d_Q)}$. For computational tractability when n is large, we subsample up to $n' = \min(5000, n)$ test instances with fixed random seed 42.

The complementarity metric extracts the block $\mathbf{C}_{HQ} = \mathbf{C}[1{:}d_H, d_H{:}]$ containing correlations between Hydra and Quant features. For each Hydra feature j, we compute its maximum absolute correlation with any Quant feature: $m_j = \max_k |\mathbf{C}_{HQ}[j,k]|$. We report the median: $\mathrm{median}(\{m_1, \ldots, m_{d_H}\})$. Low median correlation indicates feature spaces capture distinct information.

We supplement cross-correlation with canonical correlation analysis (CCA) to identify shared linear subspaces. After standardizing features, CCA finds projections $\mathbf{H}_c$ and $\mathbf{Q}_c$ that maximize correlation between corresponding dimensions. We compute $r = \min(5, d_H, d_Q, \lfloor n'/2 \rfloor)$ canonical correlations $\{\rho_1, \ldots, \rho_r\}$ where $\rho_i = |\mathrm{corr}(\mathbf{H}_{c,i}, \mathbf{Q}_{c,i})|$, sorted in descending order.

3.1.2 Prediction-level complementarity

Let $\hat{\mathbf{y}}_H, \hat{\mathbf{y}}_Q \in \{0, \ldots, c-1\}^{n_{\text{test}}}$ denote predictions from Hydra and Quant on the full test set, with ground truth $\mathbf{y} \in \{0, \ldots, c-1\}^{n_{\text{test}}}$. We define binary error vectors where $e_{H,i} = 1$ if Hydra misclassifies sample i and 0 otherwise, and similarly for $e_{Q,i}$:

$$e_{H,i} = \begin{cases} 1 & \text{if } \hat{y}_{H,i} \neq y_i \\ 0 & \text{otherwise} \end{cases}, \quad e_{Q,i} = \begin{cases} 1 & \text{if } \hat{y}_{Q,i} \neq y_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Error correlation $\rho(\mathbf{e}_H, \mathbf{e}_Q)$ quantifies whether algorithms make mistakes on the same samples. Low correlation indicates complementary error patterns. We also compute the disagreement rate (fraction of samples where algorithms predict different classes) and oracle ensemble accuracy (fraction of samples where at least one algorithm is correct):

$$D = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} [\hat{y}_{H,i} \neq \hat{y}_{Q,i}], \quad \mathrm{Acc}_{\text{oracle}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} [(\hat{y}_{H,i} = y_i) \text{ or } (\hat{y}_{Q,i} = y_i)] \quad (3)$$

where the bracket notation $[\cdot]$ evaluates to 1 if the condition is true and 0 otherwise.

The oracle ensemble represents a hypothetical perfect combiner that selects the correct prediction whenever at least one algorithm is correct, providing an upper bound *specifically for prediction-combination strategies* (voting, weighting, meta-learning from logits). This bound does not constrain feature-concatenation ensembles: a joint model trained on $[\mathbf{H} \mid \mathbf{Q}]$ can exceed oracle accuracy by learning feature interactions invisible to individual models. The oracle gain $\Delta_{\text{oracle}} = \mathrm{Acc}_{\text{oracle}} - \max(\mathrm{Acc}_H, \mathrm{Acc}_Q)$ quantifies theoretical improvement available through prediction combination.

3.2 Ensemble Design Space

Combining Hydra and Quant requires several design decisions: what representations to combine (raw features, predictions, or logits), where to apply the meta-learner, and which meta-learner architecture to use. We systematically explore this design space through six ensemble configurations that cover the primary combination strategies while maintaining computational feasibility.

3.2.1 Feature Concatenation Ensembles

The simplest combination strategy concatenates Hydra and Quant features and trains a single classifier on the joint representation: $[\mathbf{H} \mid \mathbf{Q}] \rightarrow$ Classifier, where $\mathbf{H} \in \mathbb{R}^{n \times d_H}$ represents Hydra features ($d_H \approx 10{,}000$) and $\mathbf{Q} \in \mathbb{R}^{n \times d_Q}$ represents Quant features (d_Q proportional to series length).

This approach requires careful consideration of the meta-learner. Hydra's features are individually weak: the competing kernel paradigm means each feature simply counts pattern occurrences, with discriminative power emerging from combinations of many features. Quant's interval quantiles are similarly distributional summaries rather than strong individual discriminators. A linear classifier like Ridge may struggle to exploit these heterogeneous feature spaces effectively, as it cannot learn the non-linear interactions between pattern counts and distributional statistics.

We therefore test two classifiers for the joint feature space. **Ridge (FC-Ridge)** provides a linear baseline with L2 regularization (Ridge is Hydra's default classifier), offering computational efficiency and built-in regularization for the high-dimensional feature space ($d_H + d_Q \approx 10{,}000+$). **ExtraTrees (FC-ET)** uses an ensemble of 200 extremely randomized decision trees with entropy criterion and max features set to 0.1 (ExtraTrees is Quant's default classifier; a random subset of 10% of all features are considered at each split). This non-linear ensemble can learn complex decision boundaries and feature interactions. The linear proportion of features per split (rather than $\sqrt{p}$) is critical: with large feature spaces, a sublinear number of candidate features per split could result in the classifier running out of training samples before adequately exploring the feature space (Dempster et al. 2024).

Feature concatenation has a key advantage: both Hydra and Quant features can be computed directly on the training set without information leakage concerns, as neither involves fitted model predictions at the feature extraction stage.

3.2.2 Asymmetric Stacking: Quant Features with Hydra Logits

Stacking (Wolpert 1992) provides an alternative to raw feature concatenation by using one algorithm's predictions as inputs to another. For Hydra and Quant, an asymmetric stacking approach combines Quant's features (deterministic transformations) with Hydra's logits (outputs of a fitted Ridge classifier): $[\mathbf{Q} \mid \mathbf{L}_H] \rightarrow$ Classifier, where $\mathbf{L}_H \in \mathbb{R}^{n \times c}$ represents Hydra's class probability estimates (c is number of classes). This arrangement provides a simple method to aggregate information from Hydra features (which are individually weak pattern counts) and feed these aggregated predictions in conjunction with Quant's distributional features to the final classifier.

Quant features are deterministic transformations (interval quantiles) computed directly from time series, introducing no information leakage. Hydra logits, however, come from a fitted Ridge classifier. To prevent overfitting, we generate Hydra logits via 5-fold stratified cross-validation:

OOF logit generation procedure:

$$\text{For each fold } k = 1, \ldots, 5: \quad \mathbf{L}_H^{(k)} = \text{Ridge}(\mathcal{D}_{\text{train}} \setminus F_k) \to F_k \tag{4}$$

where F_k is the held-out fold and $\mathcal{D}_{\text{train}} \setminus F_k$ denotes training on all samples except fold k. Concatenating predictions across folds yields out-of-fold logits $\mathbf{L}_H^{\text{OOF}} \in \mathbb{R}^{n \times c}$ where no prediction uses training data from its own sample.

The meta-learner trains on $\mathbf{X}_{\text{meta}} = [\mathbf{Q} \mid \mathbf{L}_H^{\text{OOF}}]$, combining Quant's raw distributional features with Hydra's aggregated pattern information. For test predictions, Hydra trains on full training data to generate $\mathbf{L}_H^{\text{test}}$.

We test two meta-learners: **Ridge (QFeat-HLogit-Ridge)** for linear combination, and **ExtraTrees (QFeat-HLogit-ET)** for non-linear combination enabling complex decision rules.

3.2.3 Symmetric Stacking: Dual Out-of-Fold Predictions

A symmetric alternative generates OOF predictions for both algorithms and trains the meta-learner on these predictions alone: $[\mathbf{L}_H^{\text{OOF}} \mid \mathbf{L}_Q^{\text{OOF}}] \to \text{ExtraTrees}$. This approach treats both algorithms identically, with neither receiving preferential treatment through raw feature access.

Both algorithms follow the same OOF generation procedure:

$$\begin{aligned} \text{Hydra: } \mathbf{L}_H^{\text{OOF}} &\in \mathbb{R}^{n \times c} \quad \text{(Ridge classifier on Hydra features)} \\ \text{Quant: } \mathbf{L}_Q^{\text{OOF}} &\in \mathbb{R}^{n \times c} \quad \text{(ExtraTrees on Quant features)} \end{aligned} \tag{5}$$

where each algorithm uses 5-fold cross-validation to produce predictions for held-out samples. The meta-learner trains on concatenated logits $[\mathbf{L}_H^{\text{OOF}} \mid \mathbf{L}_Q^{\text{OOF}}]$ (dimension $2c$), learning which algorithm to trust under which conditions based solely on their predicted class distributions.

We refer to this as **Dual-OOF-ET**. This symmetric design has theoretical appeal: it places both algorithms on equal footing and lets the meta-learner discover reliability patterns without bias toward one algorithm's feature space. However, it discards potentially useful information: the raw features that were used to generate the predictions. Whether this simplification aids or hinders meta-learning is an empirical question.

3.2.4 Weighted Averaging: Simplified CAWPE

We employ a simplified variant of Cross-validation Accuracy Weighted Probabilistic Ensemble (CAWPE) (Large et al. 2019). The original method performs 10-fold cross-validation to estimate each algorithm's competence and uses these estimates to weight

probability predictions. Our simplified approach uses training set accuracy as a proxy to reduce computational overhead.

For each algorithm j, we estimate accuracy acc_j and compute weight $w_j = \text{acc}_j^\alpha$, where $\alpha = 4$ exponentially amplifies differences in estimated competence. Final predictions combine weighted probabilities:

$$P(y = i \mid \mathbf{x}) = \frac{w_H \cdot P_H(y = i \mid \mathbf{x}) + w_Q \cdot P_Q(y = i \mid \mathbf{x})}{w_H + w_Q} \tag{6}$$

This approach has two key advantages: computational simplicity (no additional classifier training) and theoretical grounding (the exponential weighting significantly outperformed simple averaging, stacking, and ensemble selection methods across 206 datasets (Large et al. 2019)). The choice of $\alpha = 4$ balances trusting accuracy estimates (higher α puts more weight on the seemingly better algorithm) against hedging through combination (lower α weights both algorithms more equally). Our implementation details are described in Section 4.2.

3.2.5 Summary of Ensemble Configurations

Table 1 summarizes our six ensemble configurations. The design space exploration covers feature concatenation (testing linear vs. non-linear meta-learners), asymmetric stacking with OOF predictions (again with both meta-learners), symmetric stacking treating both algorithms identically, and weighted averaging without meta-learning. This systematic approach enables us to identify which combination strategies effectively exploit Hydra-Quant complementarity while maintaining computational feasibility.

Table 1 Summary of six ensemble configurations exploring the design space of combining Hydra and Quant. Specific hyperparameters for all configurations are detailed in Section 4.2.

Strategy	Inputs to Meta-Learner	Meta-Learner	OOF?
FC-Ridge	$[\mathbf{H} \mid \mathbf{Q}]$	Ridge	No
FC-ET	$[\mathbf{H} \mid \mathbf{Q}]$	ExtraTrees	No
QFeat-HLogit-Ridge	$[\mathbf{Q} \mid \mathbf{L}_H^{\text{OOF}}]$	Ridge	Yes (Hydra)
QFeat-HLogit-ET	$[\mathbf{Q} \mid \mathbf{L}_H^{\text{OOF}}]$	ExtraTrees	Yes (Hydra)
Dual-OOF-ET	$[\mathbf{L}_H^{\text{OOF}} \mid \mathbf{L}_Q^{\text{OOF}}]$	ExtraTrees	Yes (both)
Simplified CAWPE	Weighted prob. combination	None	No

4 Experimental Design

4.1 Datasets

We evaluate on 10 datasets from the MONSTER benchmark (Dempster et al. 2025a), a large-scale time series classification archive designed to address computational and evaluation challenges of modern TSC. A key motivation for using MONSTER is that existing ensemble approaches like HIVE-COTE 2.0 are computationally constrained to smaller benchmarks like UCR. They cannot realistically be applied to larger, real-world datasets due to prohibitive training times. This work investigates whether efficient ensembles can capture complementarity benefits while remaining practically applicable to larger-scale problems. The 10 datasets were selected from an initial complementarity analysis based on successful completion of feature extraction and prediction generation for both algorithms.

The selected datasets span diverse application domains: human activity recognition (UCIActivity, USCActivity, WISDM, WISDM2), vehicular and pedestrian sensors (FordChallenge, Traffic, Pedestrian), environmental monitoring (LakeIce), remote sensing (Tiselac), and audio classification (InsectSound). These 10 datasets provide coverage across univariate and multivariate problems, with training set sizes from 7,898 to 1,168,774 instances, class counts from 2 to 82, and series lengths from 23 to 600 time points. Complete dataset characteristics are provided in Appendix B.

All experiments use predefined train/test splits from the MONSTER benchmark to ensure reproducibility and fair comparison across all ensemble configurations.

4.2 Implementation

4.2.1 Base Algorithms

We employ Hydra and Quant as described in Section 2.3. For Hydra, we use $k = 8$ kernels per group, $g = 64$ groups (yielding 512 total kernels per dilation level), and random seed 42 for kernel initialization. For Quant, we use depth $d = 6$ (creating dyadic intervals at scales $\{2^0, 2^1, ..., 2^5\}$) with divisor $v = 4$ (computing $1 + \lfloor (m-1)/4 \rfloor$ quantiles per interval of length m). Quant operates on four representations: original series, smoothed first difference (moving average window=5), second difference, and discrete Fourier transform magnitude (Dempster et al. 2024).

4.2.2 Feature Concatenation Ensembles

Ridge uses L2 regularization with automatic alpha selection via cross-validation. ExtraTrees uses 200 estimators, entropy criterion, and max_features=0.1 (10% of features considered per split), following the Quant implementation (Dempster et al. 2024). This linear proportion of features prevents feature space exhaustion on high-dimensional problems.

4.2.3 Stacking Ensembles

For asymmetric and symmetric stacking, we employ 5-fold stratified cross-validation for out-of-fold prediction generation, following established TSC benchmarking protocols (Bagnall et al. 2017). The ExtraTrees meta-learners use the same configuration as feature concatenation (200 estimators, entropy criterion, max_features=0.1). Ridge meta-learners use the same cross-validated alpha selection.

4.2.4 Simplified CAWPE

We use training set accuracy as a proxy for cross-validation accuracy estimation, with exponential weighting parameter $\alpha = 4$. This simplifies the original CAWPE formulation (Large et al. 2019), which employs 10-fold cross-validation to estimate component accuracies. Our simplified approach reduces computational overhead for systematic exploration across six ensemble configurations and 10 large-scale datasets. While this may overestimate component reliability compared to proper cross-validation, the exponential weighting ($\alpha = 4$) still magnifies competence differences as specified in the original method.

4.2.5 Computational Environment

All experiments run on the M3 HPC cluster (Monash University) using Intel Xeon Gold 6548Y+ CPUs with 8 cores and 256GB RAM allocated per job. Hydra operations use GPU acceleration where available. We use Python 3.12.11 with PyTorch 2.7.1, scikit-learn 1.6.1, and NumPy 2.2.6.

4.3 Evaluation Protocol

We measure classification accuracy on held-out test sets, reporting results from single-run experiments (no repeated trials). For each dataset, we compute the ensemble gain as the difference between ensemble accuracy and the better-performing base algorithm: $\text{Gain} = \text{Acc}_{ensemble} - \max(\text{Acc}_{Hydra}, \text{Acc}_{Quant})$. Positive gains indicate the ensemble outperforms both base algorithms.

To contextualize ensemble performance relative to theoretical potential, we define oracle utilization as the ratio of actual ensemble gain to oracle gain:

$$\text{Oracle Utilization} = \frac{\text{Ensemble Gain}}{\text{Oracle Gain}} \times 100\% \tag{7}$$

Negative oracle utilization indicates the ensemble performed worse than the better base algorithm, despite theoretical potential for improvement.

We quantify computational efficiency through wall-clock training time, measured from the start of feature extraction through final model fitting. For stacking ensembles, this includes cross-validation overhead for OOF prediction generation (training each base algorithm 5 times). For CAWPE, this includes model training for both algorithms. All timing measurements represent single runs on consistent hardware, reported as time per 1000 training samples to account for dataset size differences.

For complementarity analysis, we compute feature-level metrics (median cross-correlation between Hydra and Quant feature spaces, canonical correlation analysis) and prediction-level metrics (error correlation, disagreement rate, oracle ensemble accuracy) on test data after training both algorithms on the full training set. Feature extraction for complementarity analysis uses up to 5,000 randomly sampled test instances (random seed 42) to manage memory requirements for correlation computations on datasets with very large test sets. Note that the oracle utilization metric (Equation 1) applies specifically to prediction-combination strategies; feature-concatenation ensembles can exceed the oracle bound by learning novel feature interactions.

All experimental code, ensemble implementations, and raw results are publicly available to ensure reproducibility (see Appendix F).

5 Results

5.1 Complementarity Analysis

Following the complementarity analysis framework described in Section 3.1, we measured feature-level and prediction-level complementarity between Hydra and Quant across 10 MONSTER datasets (subsampled to 5,000 instances where necessary, per Section 4.3) to validate the hypothesis that these algorithms capture orthogonal discriminatory information. Figure 1 visualizes the key complementarity patterns.

5.1.1 Feature-level complementarity

As detailed in Section 3.1, feature correlation (measured by median maximum absolute correlation between feature spaces and canonical correlation analysis) quantifies the extent to which Hydra and Quant capture overlapping versus orthogonal information. Median cross-correlation between Hydra and Quant feature representations ranged from 0.460 (FordChallenge) to 0.709 (USCActivity), with mean 0.583 ± 0.089. This moderate correlation indicates that while the feature spaces share some information, substantial complementarity exists. The lowest correlations occurred on high-dimensional multivariate data (FordChallenge: 0.460, Tiselac: 0.514), where Hydra's convolution-based patterns and Quant's quantile-based distributions capture fundamentally different signal characteristics. Higher correlations appeared on activity recognition data (USCActivity: 0.709, UCIActivity: 0.663), suggesting representational overlap when discriminatory information concentrates in common temporal features.

Canonical correlation analysis confirmed these patterns. The first canonical correlation averaged 0.997 across datasets (range: 0.987 to 1.000), indicating strong linear dependency in a single dimension, but this reflects the high dimensionality of both feature spaces rather than complete overlap. The shared information occupies low-dimensional subspaces while most feature dimensions remain independent.

5.1.2 Prediction-level complementarity

As noted in Section 3.1, prediction-level complementarity quantifies the extent to which different algorithms correctly classify given examples in complementary patterns (making errors on different samples rather than the same samples). Error correlation between Hydra and Quant predictions averaged 0.421 ± 0.181 across datasets, ranging from near-zero (UCIActivity: 0.139) to high correlation (Tiselac: 0.695). Low error correlation indicates complementary decision boundaries, where algorithms make mistakes on different samples. Disagreement rates averaged 21.9%, with substantial variation (1.0% on LakeIce to 39.8% on Traffic), reflecting dataset-dependent complementarity strength.

Oracle ensemble accuracy, measuring the fraction of test samples correctly classified by at least one algorithm, exceeded the better individual algorithm by 0.0014 to 0.1239 (mean gain: 0.0471, or 4.71% absolute improvement over best base). This quantifies the theoretical upper bound achievable through prediction-combination strategies. The largest gains appeared on multi-class problems with moderate individual accuracy (USCActivity: 12.39% gain, InsectSound: 7.73% gain, WISDM2: 6.14% gain), where algorithms exhibit different class-specific strengths. Minimal gains occurred on problems where one algorithm dominates (LakeIce: 0.14% gain, where Quant achieves 99.7% accuracy).

5.1.3 Feature concatenation exceeds oracle bound

While the oracle bound constrains prediction-combination strategies, feature-concatenation ensembles can exceed it by learning interactions invisible to individual models. Empirically, FC-ExtraTrees correctly classified 12,556 test samples (across all 10 datasets) where both Hydra and Quant failed, demonstrating that joint feature-space models can indeed surpass the prediction-combination ceiling. This oracle-exceeding rate varied by dataset, from 3.6% of "both wrong" samples (Tiselac) to 91.7% (UCIActivity), indicating that the capacity to learn novel feature interactions depends on problem characteristics.

Figure 1 summarizes these findings. The results confirm that Hydra and Quant capture complementary discriminatory information, with complementarity strength varying across problem types. Unlike previous studies that inferred complementarity from performance differences across problem types, we directly quantified complementarity through feature correlation and error analysis within each dataset.

5.2 Ensemble Performance

Table 2 presents comprehensive results comparing Hydra, Quant (two base algorithms), and the top three ensemble configurations across all 10 datasets (full results for all six ensemble configurations are provided in Appendix A). As evident from the table, ensembles achieved the highest accuracy on 7 out of 10 datasets (outperforming both base methods), with QFeat-HLogit-ET and Simplified CAWPE proving most accurate overall rather than the symmetric DualOOF-ET approach. These three
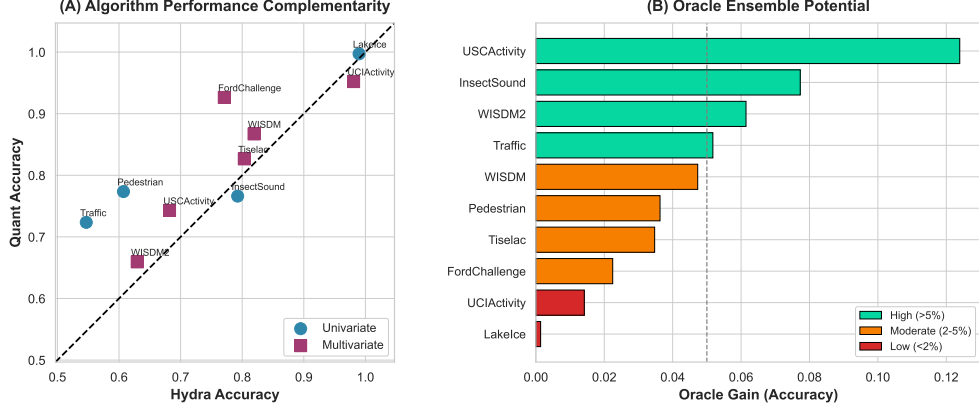
Fig. 1 Complementarity between Hydra and Quant across 10 MONSTER datasets. (A) Per-dataset accuracy comparison (univariate: blue, multivariate: orange). (B) Oracle ensemble potential (theoretical upper bound for prediction-combination strategies). Green: high potential ($> 5\%$), orange: moderate (2–5%), red: low ($< 2\%$). Feature-concatenation ensembles can exceed these bounds.

Table 2 Algorithm performance comparison across 10 MONSTER datasets. Accuracy values reported on held-out test sets. Best result per row in **bold**.

Dataset	Hydra	Quant	Best Base	QFeat+HLogit-ET	CAWPE	DualOOF-ET
FordChallenge	0.7804	0.9225	0.9225	**0.9278**	0.9236	0.9188
InsectSound	0.7826	0.7656	0.7826	0.8098	0.7936	**0.8120**
LakeIce	0.9891	0.9969	0.9969	0.9972	0.9969	**0.9973**
Pedestrian	0.6064	0.7776	0.7776	**0.7777**	0.7756	0.7731
Tiselac	0.8023	0.8243	0.8243	0.8249	0.8288	**0.8301**
Traffic	0.5446	0.7235	0.7235	0.7181	0.7254	**0.7277**
UCIActivity	0.9833	0.9579	0.9833	0.9796	0.9838	**0.9917**
USCActivity	0.6652	0.7486	0.7486	**0.7716**	0.7603	0.7369
WISDM	0.8099	0.8698	0.8698	**0.8932**	0.8897	0.8930
WISDM2	0.6304	**0.6648**	**0.6648**	0.6626	0.6594	0.6326
Mean	0.7594	0.8252	0.8294	0.8362	0.8337	0.8313

ensembles represent the strongest performers from our systematic design space exploration: QFeat-HLogit-ET (asymmetric stacking with ExtraTrees), Simplified CAWPE (weighted averaging), and DualOOF-ET (symmetric stacking).

Ridge vs ExtraTrees meta-learners. Figure 2A shows meta-learner choice significantly affects feature concatenation ensembles. ExtraTrees consistently outperforms Ridge (mean accuracy 0.827 vs 0.797), demonstrating non-linear meta-learning is essential for heterogeneous feature spaces: Ridge cannot effectively combine Hydra's pattern-counting features with Quant's distributional statistics, while ExtraTrees learns complex decision boundaries adaptively weighting different feature types.

15
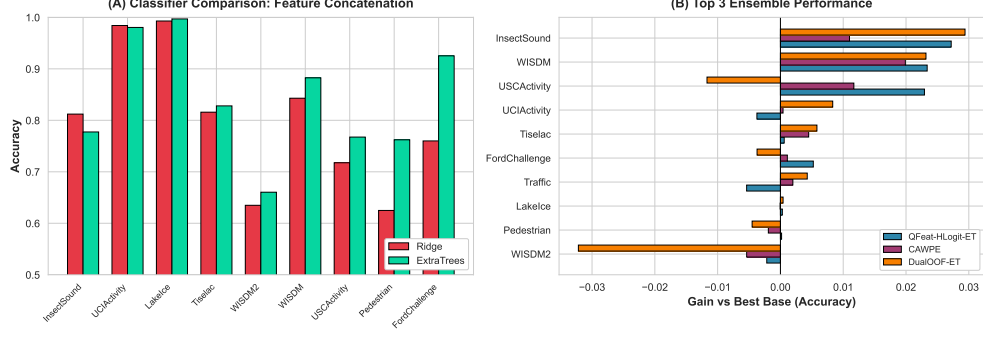
Fig. 2 Ensemble performance analysis across 10 MONSTER datasets. (A) Classifier comparison for feature concatenation strategy: Ridge (red) vs ExtraTrees (green) meta-learners across 9 datasets. ExtraTrees consistently outperforms Ridge, demonstrating the importance of non-linear meta-learning for heterogeneous feature spaces. (B) Top three ensemble performance gains relative to best base algorithm. QFeat-HLogit-ET (primary ensemble), Simplified CAWPE (weighted averaging), and DualOOF-ET (symmetric stacking) show varying effectiveness across datasets, with largest improvements on USCActivity, WISDM, and InsectSound.

Top ensemble performance. Figure 2B presents gains for the three best ensembles. QFeat-HLogit-ET achieved mean accuracy 0.836, a 0.72% improvement over best-base (0.829), succeeding on 7 of 10 datasets with largest improvements on USCActivity (+2.29pp), WISDM (+2.34pp), and InsectSound (+2.72pp). Simplified CAWPE achieved 0.834 with more consistent but smaller gains. Neither configuration universally improved over base algorithms.

DualOOF-ET, despite its symmetric design, achieved mean accuracy of 0.831. Its strength lay in consistency: it provided the largest gain on UCIActivity (+8.33pp, where other ensembles struggled), but showed mixed results elsewhere.

No hero ensemble. No single ensemble dominates across all datasets. Of the 10 datasets, QFeat-HLogit-ET ranks first on 4, Simplified CAWPE on 3, and DualOOF-ET on 1 (with base algorithms winning the remaining 2). This finding demonstrates that ensemble effectiveness is problem-dependent: the optimal combination strategy depends on the specific complementarity pattern between algorithms for each dataset, rather than one strategy being universally superior.

5.3 Predictors of Ensemble Success

We analyzed correlations between complementarity metrics and ensemble gains for QFeat-HLogit-ET (our strongest configuration). Table C (Appendix C) provides complete complementarity statistics. Figure 3 visualizes relationships between metrics and gains.

Oracle gain showed moderate positive correlation with ensemble gain (Pearson $r = 0.631$, $p = 0.050$), approaching but not reaching statistical significance. Feature complementarity (median cross-correlation) showed no significant correlation with ensemble gain ($r = 0.328$, $p = 0.36$). Dataset properties (number of classes, series length, channels, training size) also showed no significant correlations (all $|r| < 0.4$,
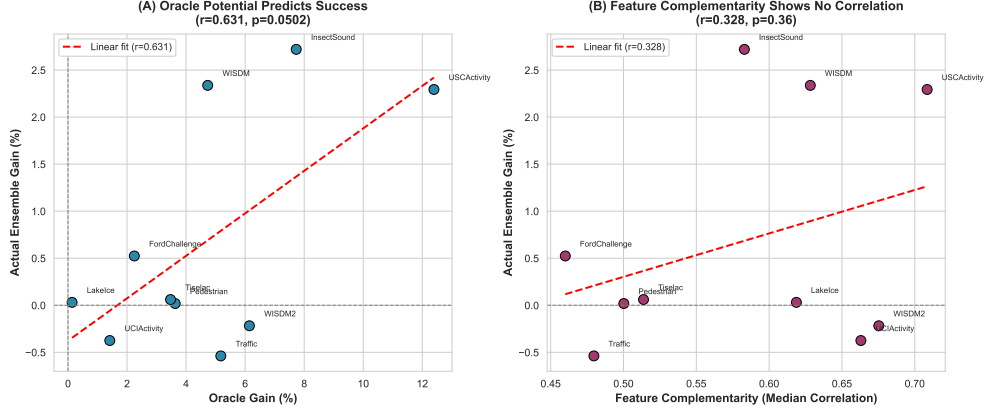
Fig. 3 Predictors of ensemble success for QFeat-HLogit-ET across 10 MONSTER datasets. (A) Oracle gain vs ensemble gain ($r = 0.631$, $p = 0.050$). (B) Feature complementarity vs ensemble gain ($r = 0.328$, $p = 0.36$).

$p > 0.1$). These correlations are exploratory; the moderate oracle gain correlation should be validated on larger dataset collections. QFeat-HLogit-ET captured only 11.0% of theoretical oracle potential on average, with substantial variation across datasets (range: –26.5% to 49.4%).

6 Discussion

Our systematic exploration of ensemble strategies for combining Hydra and Quant reveals both the potential and limitations of targeted ensemble approaches in time series classification. We achieved measurable improvements (mean 0.72% gain over best base), demonstrating that complementary algorithms from different paradigms can be successfully combined. However, the modest magnitude of gains and low oracle utilization (11.0% on average) indicate substantial room for improvement in meta-learning strategies.

6.1 Implications of Oracle Potential as Primary Predictor

The moderate correlation between oracle gain and ensemble success ($r = 0.631$, $p = 0.050$) suggests that prediction-level complementarity provides an upper bound on potential gains, though the relationship is not deterministic. This finding challenges the common practice of combining algorithms based primarily on representational diversity. While our results confirm that Hydra and Quant do exhibit complementary features (median cross-correlation 0.46–0.71 across datasets), this feature-level complementarity does not predict ensemble gains ($r = 0.328$, $p = 0.36$).

The oracle bound constrains only prediction-combination strategies, not feature-concatenation approaches. Our empirical results (Section 5.1) demonstrate that FC-ExtraTrees exceeded the oracle bound on all 10 datasets by correctly predicting 12,556 samples where both base algorithms failed. This confirms the theoretical prediction

17

that joint feature-space models can learn interactions invisible to individual classifiers, though the magnitude of oracle-exceeding gains remained modest (0.21–3.40% of test samples across datasets).

The practical implication is to shift ensemble design focus from "What different representations do these algorithms use?" to "Do these algorithms make mistakes on different samples?" The oracle analysis provides a simple diagnostic: if oracle gain is small (for example, LakeIce with 0.14% oracle gain), ensemble methods are unlikely to provide substantial improvements regardless of how complementary the features appear. Conversely, high oracle gain (USCActivity with 12.4%) indicates fertile ground for ensemble methods, even if feature representations show moderate correlation.

This principle extends beyond Hydra-Quant combinations. For any pair of candidate algorithms, computing oracle accuracy on a validation set provides an upper bound estimate before committing computational resources to training complex meta-learners or exploring ensemble architectures. When oracle potential is low, practitioner effort may be better spent on algorithm selection or hyperparameter tuning rather than ensemble construction.

6.2 The Meta-Learning Optimization Gap

Our ensembles captured only 11.0% of theoretical oracle potential, indicating that current meta-learning strategies are far from optimal. This raises a fundamental question: why do meta-learners struggle to exploit complementarity that oracle analysis shows exists?

We identify three potential explanations. First, **insufficient meta-learning signal**: the meta-learner trains on n OOF predictions (where n is training set size), which may be inadequate for learning reliable combination rules, particularly for datasets with many classes where the prediction space is high-dimensional. Second, **information bottleneck**: the $2c$-dimensional logit space (for c classes) may not encode sufficient information about when each algorithm should be trusted. The confidence scores reflect how certain each algorithm is, but not the problem characteristics that determine reliability. Third, **meta-learner capacity**: ExtraTrees may lack the capacity to learn complex conditional logic like "trust Hydra on long-series problems with periodic patterns, but trust Quant on short-series problems with distributional shifts."

Addressing these limitations suggests several research directions. **Richer meta-features**: Rather than using only logits, meta-learners could incorporate problem descriptors (series length, number of classes, distribution statistics) to learn context-dependent combination rules. **Instance-level features**: Computing simple statistics on the input time series (mean, variance, autocorrelation) and concatenating with logits may help the meta-learner identify when each algorithm should be trusted. **Metalearning architectures**: Neural meta-learners or gradient-boosted trees might better capture complex combination logic than ExtraTrees. **Multi-level stacking**: Training multiple meta-learner layers, where higher levels learn when to trust lower-level predictions, could iteratively refine combination strategies.

However, these directions trade increased complexity for potentially marginal gains. Given that our simplest successful configuration (QFeat-HLogit-ET) achieves only 11.0% oracle utilization with modest actual gains (0.72%), the question becomes:

is the computational overhead of more sophisticated meta-learning justified? For practitioners working with limited computational budgets, focusing on algorithm selection and hyperparameter tuning may yield better returns than elaborate ensemble architectures.

6.3 No Universal Ensemble Strategy

No single ensemble configuration dominates across all datasets. QFeat-HLogit-ET ranks first on 4 of 10 datasets, CAWPE on 3, DualOOF-ET on 1, with base algorithms winning 2. This heterogeneity mirrors the broader TSC principle that "no one representation will dominate" (Bagnall et al. 2017), extended to the meta-learning level. Ensemble selection is itself a meta-learning problem: given a dataset, which combination strategy will perform best? Our experiments identified no reliable predictors, as dataset properties showed no significant correlations with ensemble effectiveness.

For practitioners, this implies cross-validation over ensemble configurations may be necessary, though this multiplies computational overhead. Had we tested only one configuration, conclusions about Hydra-Quant complementarity would depend on which strategy was chosen. The finding that ExtraTrees universally outperforms Ridge for feature concatenation, while different high-level strategies excel on different datasets, suggests meta-learner choice and combination strategy operate at different design space levels.

6.4 Computational Cost Considerations

Our strongest ensemble (QFeat-HLogit-ET) provides a mean 0.72% accuracy improvement while increasing training time by approximately 2.5 times (from Quant alone) due to 5-fold cross-validation overhead. For practitioners, this trade-off may be unfavorable in many contexts. If computational budget is limited, running Quant (which individually outperforms Hydra on 8 of 10 datasets) may be preferable to investing resources in ensemble construction.

The cost-benefit analysis depends on problem context. For applications where small accuracy improvements are valuable (medical diagnosis, financial prediction), the 0.72% gain may justify increased computational cost. For rapid prototyping or large-scale benchmarking studies, the computational overhead may be prohibitive. Figure E (Appendix E) shows that Quant and Hydra provide the best efficiency-accuracy trade-offs, with ensembles falling in the high-cost, modest-benefit region.

This suggests a staged approach: begin with individual base algorithms to establish baselines, compute oracle gain to assess ensemble potential, and only if oracle gain exceeds a threshold (for example, 5%) invest in ensemble construction. This avoids wasted effort on datasets where theoretical potential is minimal.

6.5 Limitations

Our study has four primary limitations that future research should address.

Simplified CAWPE implementation. Our CAWPE variant uses training set accuracy rather than 10-fold cross-validation for component weighting (Large et al. 2019). This simplification may overestimate component reliability, particularly on

datasets where training accuracy substantially exceeds generalization accuracy. The original CAWPE achieved 1.19% gain on the UCR archive (Large et al. 2019, Figure 8); our simplified version achieved 0.5% mean gain on MONSTER. Future work should validate findings with proper cross-validation-based weighting to determine whether low performance reflects methodological simplification or fundamental limitations of weighted averaging for Hydra-Quant combinations.

Statistical power. With 10 datasets, correlation analyses have limited power (47% power to detect $r = 0.6$ at $\alpha = 0.05$). The oracle gain correlation ($r = 0.631$, $p = 0.050$) approaches but does not reach significance after multiple comparison correction (Bonferroni-adjusted $\alpha = 0.017$ for three tests). These correlations should be interpreted as exploratory hypothesis generation rather than confirmatory findings. Validation on the full 29-dataset MONSTER archive would strengthen conclusions about predictor reliability and potentially reveal dataset-specific moderators of ensemble success.

Algorithm scope. Our analysis focuses on two specific algorithms (Hydra and Quant) representing convolutional and interval paradigms. While the complementarity analysis framework generalizes, specific findings (QFeat-HLogit-ET as optimal configuration, 11% oracle utilization, no correlation between feature complementarity and gains) may not transfer to other algorithm pairs. Validation with additional efficient combinations (MultiROCKET + Quant, WEASEL + Hydra, InceptionTime + DrCIF) would test generalizability and identify whether low oracle utilization reflects Hydra-Quant-specific issues or fundamental meta-learning challenges.

6.6 Future Directions

Several research directions emerge. First, improving meta-learning strategies through richer meta-features (problem descriptors: series length, class count, distributional statistics) or instance-level features (time series statistics concatenated with logits) may increase oracle utilization beyond 11%. More sophisticated architectures (neural meta-learners, gradient-boosted trees, multi-level stacking) might better capture complex combination logic than ExtraTrees.

Second, validating findings on the full 29-dataset MONSTER archive would clarify how dataset scale affects combination strategies and whether patterns differ between UCR (small, curated) and MONSTER (large, real-world). Finally, developing predictive models for ensemble selection would automate the currently manual choice of which configuration to use for a given dataset.

Despite these limitations, our work provides the most comprehensive analysis to date of targeted ensemble strategies for modern efficient TSC algorithms, demonstrating both the potential and current limitations of this approach.

7 Conclusion

This work systematically investigated whether targeted ensemble strategies can capture the accuracy benefits of comprehensive multi-paradigm combination while maintaining computational efficiency for large-scale time series classification. We

combined two efficient algorithms from distinct paradigms (Hydra's competing convolutional kernels and Quant's hierarchical interval quantiles) across six ensemble configurations, evaluating performance on 10 large-scale MONSTER datasets with training sets ranging from 7,898 to 1,168,774 instances.

Our findings demonstrate that complementary efficient algorithms can be successfully combined to achieve modest but consistent improvements. The strongest configuration, QFeat-HLogit-ET (asymmetric stacking with ExtraTrees meta-learner), achieved mean accuracy of 0.836 compared to 0.829 for the best base algorithm, representing a 0.72% improvement while improving on 7 of 10 datasets. Feature concatenation with non-linear meta-learning proved essential: ExtraTrees achieved 0.842 mean accuracy versus 0.798 for Ridge regression on the same concatenated features, demonstrating that linear models underfit on these larger problems. Among combination strategies, no single approach dominated. QFeat-HLogit-ET ranked first on 4 datasets, Simplified CAWPE on 3, and DualOOF-ET on 1, mirroring the broader TSC principle that algorithm effectiveness is problem-dependent.

However, these gains fall far short of theoretical potential. Oracle analysis revealed that perfect prediction combination could achieve 4.71% mean improvement over the better base algorithm, yet actual prediction-combination ensembles captured only 11.0% of this potential on average. This optimization gap indicates that current meta-learning strategies struggle to reliably identify when to trust each algorithm's predictions. The moderate correlation between oracle gain and ensemble success ($r = 0.631$, $p = 0.050$) approaches but does not reach statistical significance, showing considerable scatter that suggests factors beyond oracle potential (such as meta-learner capacity and feature interactions) play crucial roles. Contrary to expectations, feature-level complementarity showed no significant correlation with ensemble gains ($r = 0.328$, $p = 0.36$), revealing that complementary representations alone do not guarantee success.

These findings carry practical implications for ensemble design in time series classification. First, practitioners should assess prediction-level complementarity (oracle potential) before investing in complex meta-learning architectures, as this sets an upper bound on achievable gains. When one algorithm dominates (such as Quant achieving 99.7% on LakeIce), even highly complementary features yield minimal ensemble improvements. Second, meta-learner choice matters: non-linear models like ExtraTrees significantly outperform linear approaches when combining heterogeneous features from different paradigms. Third, the low oracle utilization (11%) indicates that ensemble gains could potentially be doubled or tripled through improved meta-learning strategies, though achieving this requires solving fundamental challenges in learning when to trust each algorithm.

This work has limitations detailed in Section 6.5: simplified CAWPE implementation, limited statistical power (10 datasets), focus on two algorithms, and computational constraints. Future work (Section 6.6) should improve meta-learning through richer meta-features, instance-level feature augmentation, or more sophisticated architectures, and validate findings on the full 29-dataset MONSTER archive.

The central insight from this work is that the challenge in ensemble time series classification has shifted from ensuring algorithms are different to learning how to

21

combine them effectively. Hydra and Quant exhibit clear complementarity: moderate feature correlation (0.46 to 0.71), substantial error independence (average error correlation 0.421), and meaningful oracle potential (mean 4.71%). Yet current meta-learning captures only 11% of this potential. This gap represents both a limitation of current approaches and an opportunity for future research. As time series classification confronts increasingly large-scale real-world problems, developing efficient ensembles that effectively exploit complementarity between fast algorithms becomes increasingly important for practical applications.

Declarations

Ethics approval: This research uses only publicly available datasets and involves no human participants, personally identifiable data, or animal subjects. Ethics approval was therefore not required.

Data and code availability: All datasets, code implementations, and experimental results are publicly available as detailed in Appendix F.

Competing interests: The author declares no competing interests.

Appendix A   Complete Algorithm Comparison

Table A presents complete results across all ensemble configurations.

Table A Complete algorithm comparison across all ensemble strategies. Accuracy on held-out test sets. Missing H-U (Hydra-Univariate) entries correspond to multivariate datasets; missing H-M (Hydra-Multivariate) entries correspond to univariate datasets. FC-R on Traffic timed out due to dataset size and algorithm complexity.

Dataset	Q	H-U	H-M	FC-R	FC-ET	QH-R	QH-ET	DO-ET	CAWPE
FordChallenge	0.923	—	0.780	0.760	0.925	0.816	**0.928**	0.919	0.924
InsectSound	0.766	0.783	—	**0.812**	0.777	0.805	0.810	**0.812**	0.794
LakeIce	0.997	0.989	—	0.993	0.997	0.992	0.997	**0.997**	0.997
Pedestrian	0.778	0.606	—	0.625	0.762	0.612	**0.778**	0.773	0.776
Tiselac	0.824	—	0.802	0.816	0.828	0.821	0.825	**0.830**	0.829
Traffic	0.723	0.545	—	—	0.683	0.539	0.718	**0.728**	0.725
UCIActivity	0.958	—	0.983	0.984	0.980	0.986	0.980	**0.992**	0.984
USCActivity	0.749	—	0.665	0.718	0.767	0.733	**0.772**	0.737	0.760
WISDM	0.870	—	0.810	0.843	0.883	0.799	**0.893**	0.893	0.890
WISDM2	**0.665**	—	0.630	0.635	0.660	0.633	0.663	0.633	0.659
Mean	0.825	0.731	0.779	0.798	0.826	0.773	0.836	0.831	0.834

Appendix B   Dataset Characteristics

Table B summarizes characteristics of the 10 MONSTER benchmark datasets used in this study.

Table B Characteristics of the 10 MONSTER benchmark datasets used in this study.

Dataset	Train	Test	Channels	Length	Classes
FordChallenge	29,003	7,254	30	40	2
InsectSound	40,000	10,000	1	600	10
LakeIce	103,424	25,856	1	161	3
Pedestrian	151,696	37,925	1	24	82
Tiselac	79,907	19,780	10	23	9
Traffic	1,168,774	292,194	1	24	7
UCIActivity	7,898	2,401	9	128	6
USCActivity	43,060	13,168	6	100	12
WISDM2	98,094	50,940	3	100	6
WISDM	11,989	5,177	3	100	6

Appendix C   Complementarity Metrics

Table C presents complete complementarity statistics and ensemble performance.
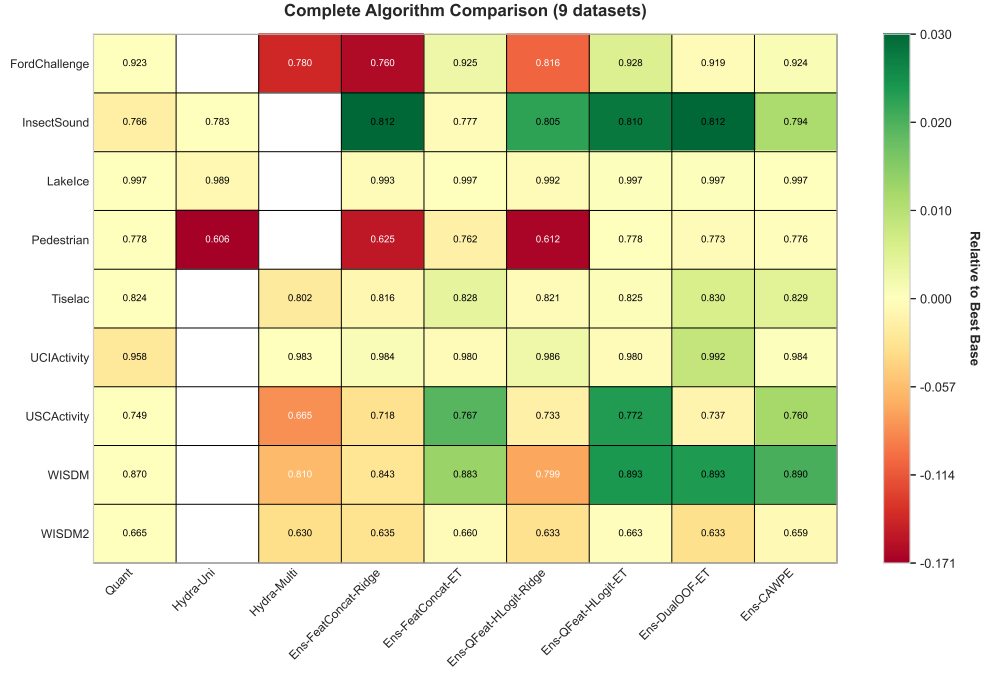
Table C Complementarity metrics and ensemble performance. Feat Corr: median feature correlation (lower = more complementary). Err Corr: error correlation. Oracle Gain: theoretical upper bound. Ensemble gains show improvement over best base algorithm.

Dataset	Feat Corr	Err Corr	Oracle	QFeat-HL	CAWPE	DualOOF
InsectSound	0.583	0.476	0.0773	+0.0272	+0.0110	+0.0294
WISDM	0.628	0.471	0.0473	+0.0234	+0.0199	+0.0232
USCActivity	0.709	0.253	0.1239	+0.0229	+0.0117	-0.0117
UCIActivity	0.663	0.139	0.0142	-0.0037	+0.0004	+0.0083
Tiselac	0.514	0.695	0.0347	+0.0006	+0.0045	+0.0058
FordChallenge	0.460	0.312	0.0225	+0.0052	+0.0011	-0.0037
Traffic	0.480	0.446	0.0518	-0.0054	+0.0020	+0.0043
LakeIce	0.619	0.256	0.0014	+0.0003	-0.0000	+0.0004
Pedestrian	0.500	0.495	0.0363	+0.0002	-0.0019	-0.0045
WISDM2	0.675	0.668	0.0614	-0.0022	-0.0054	-0.0322
Mean	0.583	0.421	0.0471	+0.0069	+0.0043	+0.0019

Table C shows substantial variation in feature-level complementarity (Feat Corr), prediction-level complementarity (Err Corr), oracle potential (Oracle Gain), and actual ensemble improvements across datasets. InsectSound, WISDM, and USCActivity exhibit the highest oracle gains (7.7%, 4.7%, 12.4% respectively) and correspondingly achieve the largest ensemble improvements. This pattern demonstrates that oracle potential (the theoretical ceiling) predicts ensemble success better than feature-level complementarity measures.

Appendix D Complete Algorithm Comparison

Figure D shows the complete algorithm comparison heatmap across all 9 datasets (Traffic excluded due to missing ensemble configuration).

Complete Algorithm Comparison (9 datasets)

	Quant	Hydra-Uni	Hydra-Multi	Ens-FeatConcat-Ridge	Ens-FeatConcat-ET	Ens-QFeat-HLogit-Ridge	Ens-QFeat-HLogit-ET	Ens-DualOOF-ET	Ens-CAWPE
FordChallenge	0.923		0.780	0.760	0.925	0.816	0.928	0.919	0.924
InsectSound	0.766	0.783		0.812	0.777	0.805	0.810	0.812	0.794
LakeIce	0.997	0.989		0.993	0.997	0.992	0.997	0.997	0.997
Pedestrian	0.778	0.606		0.625	0.762	0.612	0.778	0.773	0.776
Tiselac	0.824		0.802	0.816	0.828	0.821	0.825	0.830	0.829
UCIActivity	0.958		0.983	0.984	0.980	0.986	0.980	0.992	0.984
USCActivity	0.749		0.665	0.718	0.767	0.733	0.772	0.737	0.760
WISDM	0.870		0.810	0.843	0.883	0.799	0.893	0.893	0.890
WISDM2	0.665		0.630	0.635	0.660	0.633	0.663	0.633	0.659

Relative to Best Base: 0.030, 0.020, 0.010, 0.000, -0.057, -0.114, -0.171

Fig. D Complete algorithm comparison across 9 MONSTER datasets (Traffic excluded). Colors indicate performance relative to best base algorithm per dataset. Green cells show improvements, red shows degradation, yellow shows comparable performance. No single ensemble dominates across all datasets, validating the need for problem-specific ensemble selection.

Appendix E Computational Cost Analysis

Figure E presents computational cost analysis across all algorithms.
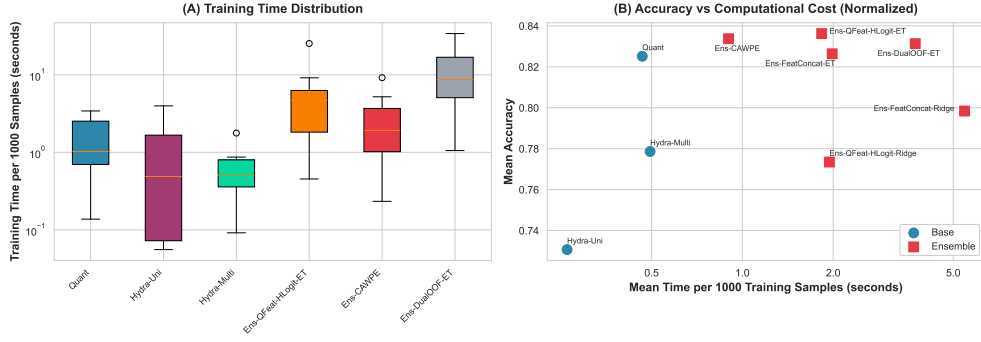
Fig. E Computational cost analysis. (A) Training time distribution per algorithm showing median (orange line) and quartiles. Ensembles with OOF prediction generation (QFeat-HLogit-ET, CAWPE, DualOOF-ET) incur 5-fold cross-validation overhead. (B) Accuracy vs normalized computational cost (time per 1000 training samples). Normalization accounts for dataset size differences. Quant and Hydra-Multi offer best efficiency, while ensembles trade computational cost for marginal accuracy gains.

Appendix F   Code Availability

All code for this research is publicly available to support reproducibility and facilitate future research in ensemble time series classification.

F.1   Algorithm Implementations

Hydra. We use the GPU-accelerated batch-processing implementation from the AALTD 2024 repository (Dempster et al. 2025b), which extends the original Hydra algorithm for large-scale datasets. The code is available at:

https://github.com/urav06/hydra/
https://github.com/urav06/aaltd2024/

Quant. Our fork restructures the original Quant implementation for improved readability and modularity while preserving the core algorithm. The code is available at:

https://github.com/urav06/quant/

Modifications to Quant are documented in the commit history and README files.

F.2   Research Repository

The complete research repository containing all ensemble implementations, experimental code, dataset handling utilities, and analysis scripts is available at:

https://github.com/urav06/research

This repository includes:

- **Ensemble implementations** (`tsckit/ensembles/`): All six ensemble configurations evaluated in this work (FC-Ridge, FC-ET, QFeat-HLogit-Ridge, QFeat-HLogit-ET, Dual-OOF-ET, Simplified CAWPE)
- **Experimental scripts** (`experiments/`): Scripts for benchmark evaluation and complementarity analysis
- **Analysis code** (`experiments/results/analysis/`): Scripts to generate all figures and tables in this paper
- **TSCKIT package** (`tsckit/`): Unified interface for running experiments across different algorithm implementations
- **Raw results** (`experiments/results/`): Complete experimental results including predictions, metrics, and timing data

All experiments were conducted using Python 3.12 with package versions specified in `requirements.txt`. Random seeds (42) are documented in experimental scripts to ensure reproducibility.

References

Bagnall A, Lines J, Bostrom A, et al (2017) The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery 31(3):606–660. https://doi.org/10.1007/s10618-016-0483-9, URL http://link.springer.com/10.1007/s10618-016-0483-9

Dempster A, Petitjean F, Webb GI (2020) Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data mining and knowledge discovery 34(5):1454–1495

Dempster A, Schmidt DF, Webb GI (2023) Hydra: competing convolutional kernels for fast and accurate time series classification. Data Mining and Knowledge Discovery 37(5):1779–1805. https://doi.org/10.1007/s10618-023-00939-3, URL https://link.springer.com/10.1007/s10618-023-00939-3

Dempster A, Schmidt DF, Webb GI (2024) Quant: a minimalist interval method for time series classification. Data Mining and Knowledge Discovery 38(4):2377–2402. https://doi.org/10.1007/s10618-024-01036-9, URL https://link.springer.com/10.1007/s10618-024-01036-9

Dempster A, Foumani NM, Tan CW, et al (2025a) Monster: Monash scalable time series evaluation repository. URL https://arxiv.org/abs/2502.15122, arXiv:2502.15122

Dempster A, Tan CW, Miller L, et al (2025b) Highly scalable time series classification for very large datasets. In: Advanced Analytics and Learning on Temporal Data, Lecture Notes in Computer Science, vol 15433. Springer, Switzerland, p 80–95

Large J, Lines J, Bagnall A (2019) A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates. Data mining and knowledge discovery 33(6):1674–1709

Middlehurst M, Large J, Flynn M, et al (2021) HIVE-COTE 2.0: a new meta ensemble for time series classification. Machine Learning 110(11-12):3211–3243. https://doi.org/10.1007/s10994-021-06057-9, URL https://link.springer.com/10.1007/s10994-021-06057-9

Middlehurst M, Schäfer P, Bagnall A (2024) Bake off redux: a review and experimental evaluation of recent time series classification algorithms. Data Mining and Knowledge Discovery 38(4):1958–2031. https://doi.org/10.1007/s10618-024-01022-1, URL https://link.springer.com/10.1007/s10618-024-01022-1

Wolpert DH (1992) Stacked generalization. Neural networks 5(2):241–259