RocketPFN: Accurate Time Series Classification via In-Context Learning

Anonymous authors
Paper under double-blind review

Abstract

We introduce RocketPFN, a training-free pipeline for time series classification that combines random convolutional feature extraction (Rocket) with in-context classification via a pretrained tabular foundation model (TabPFN v2.5). On 92 UCR datasets (30-resample protocol), RocketPFN matches HC2, the strongest published method on the archive, in mean accuracy (both 0.900, Wilcoxon $p = 0.50$), with no training on the target data and a median inference time of 30 seconds per fold. It also significantly outperforms every individual classifier in the HC2 ensemble. On UEA (20 datasets) the difference is likewise not statistically significant. A separate comparison concerns TSC foundation models: when paired with the same downstream classifier, MOMENT, Mantis, and MantisV2 are all significantly outperformed by RocketPFN using fewer extracted features and no learned parameters ($p < 0.001$ in each case). This holds even when the encoders were pretrained on corpora that include the UCR training samples. We propose this two-stage pipeline as a reference point for evaluating zero-shot TSC foundation models.

1 Introduction

Time series classification is a core problem across science and engineering. Detecting seizures from EEG (Chaovalitwongse et al., 2007), recognizing human activity from wearable sensors (Lara and Labrador, 2013), monitoring industrial equipment for fault diagnosis (Lei et al., 2020), and classifying seismic events (Arul and Kareem, 2021) all reduce to the same fundamental task: assigning a discrete label to a sequence of measurements. The UCR (Dau et al., 2019) and UEA (Bagnall et al., 2018) archives have provided a standardized benchmark of over 150 datasets spanning univariate and multivariate settings, and have driven steady, measurable progress on the problem over two decades.

The strongest published method on these archives is HIVE-COTE 2.0 (HC2) (Middlehurst et al., 2021), a meta-ensemble of four specialized classifiers, each itself an ensemble, covering shapelets, dictionaries, interval-based features, and convolutions, combined via a weighted probabilistic controller. HC2 achieves the highest mean accuracy on UCR and UEA, but at a steep computational cost, with training times of hours to days per dataset (Middlehurst et al., 2021).

A separate family of methods reaches near-HC2 accuracy at a small fraction of its cost. Rocket (Dempster et al., 2020) and its successors, map each time series to a fixed-length vector of convolutional summaries computed from randomly sampled kernels. These vectors are deliberately overcomplete: default Rocket uses 10,000 kernels to produce 20,000 features, often exceeding the number of training examples. This makes a strongly regularised linear classifier the natural choice for extracting signal from the random feature space. Despite their simplicity, the accuracy gap between this family and HC2 on UCR is small (Middlehurst et al., 2024b).

On a separate line, TSC foundation models (Goswami et al., 2024; Feofanov et al., 2025; 2026) invest substantial design effort, pretraining data, and compute in the feature extraction step. The motivation is to learn embeddings of moderate dimensionality (typically 256 to 1,024) that improve quality and compactness, enabling more expressive non-linear downstream classifiers and, ultimately, higher accuracy. Used as frozen feature extractors, these models do not currently match HC2 or the random-kernel family on UCR or UEA.

Meanwhile, a different class of classifiers has matured in the tabular machine learning community. TabPFN (Hollmann et al., 2023; 2025) is a Prior-data Fitted Network (Müller et al., 2022), pretrained on millions of synthetic datasets to approximate Bayesian inference via in-context learning: it receives a labeled training set and unlabeled test points as input and predicts class probabilities in a single forward pass, with no gradient updates on the target data. The version we build on, TabPFN v2.5 (Prior Labs Team, 2025), scales to 50,000 samples and 2,000 features, outperforming tuned gradient-boosted trees on standard tabular benchmarks. Crucially, like Rocket, TabPFN requires no fitting on the target task, making it well suited to small-data regimes, but unlike the linear classifiers typically used after Rocket, it can represent non-linear decision rules.

This raises a natural question. Can a training-free pipeline, combining random convolutional features with in-context tabular classification, reach the accuracy of the methods that dominate these archives?

We answer this question experimentally. We introduce RocketPFN, a two-stage, training-free pipeline: random convolutional feature extraction (à la Rocket) followed by in-context classification via TabPFN v2.5. The design is constrained by a practical mismatch—Rocket produces 20,000 features from 10,000 kernels, while TabPFN is designed for up to 2,000—which we resolve by splitting kernels into G groups of 1,000 and ensembling the resulting probability estimates. No parameters are fitted to the target data at any stage.

On 92 datasets of the UCR archive (30-resample protocol), RocketPFN (G=10) achieves the same mean accuracy as HC2 (both 0.900, Wilcoxon $p = 0.50$) (Wilcoxon, 1945), while requiring only a median of 30 seconds per fold on a GPU. It significantly outperforms every individual classifier in the HC2 ensemble, as well as Rocket, InceptionTime, HC1, and TS-CHIEF (Fawaz et al., 2020; Lines et al., 2018; Shifaz et al., 2020). Even at one-tenth the kernel budget (G=1, 1,000 kernels), RocketPFN already significantly outperforms all four HC2 components and Rocket on UCR.

A second comparison concerns the zero-shot use of TSC foundation models. When MOMENT (Goswami et al., 2024), Mantis (Feofanov et al., 2025), and MantisV2 (Feofanov et al., 2026) are used as zero-shot feature extractors and paired with the same TabPFN downstream classifier, all three are significantly outperformed by RocketPFN at G=1 ($p < 0.001$ in each case, 103 UCR datasets). This holds even when the encoders were pretrained on corpora that include the UCR training samples. It also holds when RocketPFN is restricted to k=100 kernels and 200 features, fewer than the embedding dimension of any of the three encoders, ruling out dimensionality as a confound.

We propose this two-stage pipeline—random convolutional features classified in context by a pretrained tabular foundation model—as a reference point for zero-shot TSC foundation models. The pretraining cost and design complexity required to learn low-dimensional embeddings should be justified by concrete conditions under which those embeddings outperform this training-free baseline.

Our main contributions are as follows:

1. **A training-free pipeline that matches the state of the art.** RocketPFN (G=10) achieves the same mean accuracy as HC2 on UCR (both 0.900, $p = 0.50$) and is not significantly different on UEA, with no parameter fitting on the target data and a median wall-clock time of 30 seconds per fold (Sections 4.4–4.6).

2. **In-context learning outperforms ridge regression on random convolutional features.** At the same 10,000-kernel budget, replacing ridge regression with TabPFN yields a significant accuracy gain on UCR, confirming that the classifier is a non-trivial contributor (Section 4.4).

3. **Temporal encoding is necessary.** TabPFN applied to raw flattened time series fails completely on trivially solvable tasks (Section 4.2).

4. **Random convolutional features as a reference point for zero-shot TSC foundation models.** RocketPFN (G=1) significantly outperforms MOMENT, Mantis, and MantisV2—each paired with the same downstream classifier—on 103 UCR datasets, including models pretrained on the UCR training splits themselves (Section 4.5).

2 Related Work

A time series classification dataset consists of n instances $\{(\mathbf{X}_i, y_i)\}_{i=1}^n$, where each $\mathbf{X}_i \in \mathbb{R}^{m \times t}$ is a (possibly multivariate) series with m channels and t time steps, and $y_i \in \{1, \ldots, C\}$ is a class label. When $m = 1$ the problem is univariate; the UEA archive focuses on the multivariate case, where $m > 1$.

Random convolutional features. Rocket (Dempster et al., 2020) transforms each time series into a fixed-length feature vector by applying a large number of random convolutional kernels. Each kernel has a randomly sampled length $l \in \{7, 9, 11\}$, weights drawn from $\mathcal{N}(0, 1)$ and mean-centered, bias from $\mathcal{U}(-1, 1)$, and dilation $d = \lfloor 2^x \rfloor$ with $x \sim \mathcal{U}(0, A)$, where $A = \log_2 \frac{t-1}{l-1}$, so that the effective receptive field can span the full series length. Padding is applied with probability 1/2; stride is always one. Each kernel is convolved with the input to produce a feature map, from which two summary statistics are extracted: the global maximum and the proportion of positive values (PPV). With k kernels this yields a $2k$-dimensional feature vector per instance. The default configuration uses $k = 10{,}000$ kernels and feeds the resulting 20,000 features to a ridge regression classifier.

For multivariate series ($m > 1$), each kernel is assigned a randomly selected subset of K channels, where $K = \lfloor 2^u \rfloor$ with $u \sim \mathcal{U}(0, \log_2(\min(m, l) + 1))$, so most kernels focus on a small number of channels. Each selected channel receives its own weight vector of length l, drawn independently from $\mathcal{N}(0, 1)$ and mean-centered—yielding a kernel that acts as a proper multi-channel convolution restricted to the selected channels. At each position, the outputs across all selected channels are summed into a single scalar; max and PPV are then computed over this summed signal. Each kernel therefore still produces exactly two features regardless of the number of channels or the size of the selected subset.

The key insight behind Rocket is that while any single random kernel captures only a weak signal, the aggregate of thousands of diverse kernels—spanning different lengths, dilations, and weight patterns—produces a rich, discriminative representation. Dilation is particularly important: it allows short kernels to detect patterns at multiple temporal scales, from fine-grained local motifs to broad trends.

In-context learning for tabular data. TabPFN (Hollmann et al., 2023; 2025) is a Prior-data Fitted Network (Müller et al., 2022)—a transformer pre-trained on synthetic tabular datasets sampled from structural causal models. At inference time it receives the entire labeled training set and unlabeled test points as context, and outputs predicted class probabilities in a single forward pass approximating the Bayesian posterior predictive distribution $p(y \mid \mathbf{x}, \mathcal{D}_{\text{train}})$ without any gradient updates on the target data. TabPFN v2.5 (Prior Labs Team, 2025) extends this to 50,000 training samples and is optimized for up to 2,000 features, using a deeper architecture (24 layers for classification), larger feature group size, and 64 learned thinking rows that provide additional computational capacity. Predictions are aggregated across an internal ensemble of $e = 8$ estimators, each applying a distinct combination of preprocessing transformations (robust scaling, soft clipping, quantile normalization, standard scaling), random column and label-index shuffling, and a random subsample of 500 features; some estimators additionally augment the feature set with SVD components. Concurrent with this work, TabPFN v3 (Grinsztajn et al., 2026) was released, raising the supported number of samples, features, and classes (which extends coverage of the UCR and UEA archives), alongside changes to the architecture and training prior. We use TabPFN v2.5 throughout; a brief assessment of RocketPFN under TabPFN v3 is reported in Appendix C.

Time series classification foundation models. An alternative to hand-crafted features is to learn temporal representations via large-scale pretraining. MOMENT (Goswami et al., 2024), Mantis (Feofanov et al., 2025), and MantisV2 (Feofanov et al., 2026) are time series foundation models that can be used as zero-shot feature extractors for classification: each is pretrained on real or synthetic time series data and used as a zero-shot feature extractor, mapping each series through a frozen encoder to a fixed-length embedding (256–1 024 dimensions) that is then passed to a downstream classifier. MOMENT is pretrained on a large public corpus that includes the UCR and UEA training splits, among others, using a masked reconstruction objective. Mantis is pretrained on a union of public time series collections that includes UCR and UEA training splits, using a contrastive learning objective. Neither model is exposed to test split or labels at any stage. MantisV2 uses synthetic data generated via CauKer (Xie et al., 2026).

State-of-the-art ensembles. HC2 (Middlehurst et al., 2021) is a meta-ensemble of four classifiers, each targeting a different representation domain: the Shapelet Transform Classifier (STC), the Temporal Dictionary Ensemble (TDE), the Diverse Representation Canonical Interval Forest (DrCIF), and Arsenal—itself an ensemble of Rocket classifiers. A weighted probabilistic controller (CAWPE) combines their predictions. HC2 is currently the most accurate method on the UCR and UEA archives, significantly outperforming all individual classifiers including Rocket, InceptionTime, and TS-CHIEF.

3 Method

RocketPFN is a two-stage pipeline: random convolutional feature extraction followed by in-context classification.

Given a dataset $\{(\mathbf{X}_i, y_i)\}_{i=1}^{n}$ with $\mathbf{X}_i \in \mathbb{R}^{m \times t}$, each instance is first z-normalised along the time axis to zero mean and unit variance—the standard preprocessing recommended for Rocket (Dempster et al., 2020) and applied by default in the aeon toolkit (Middlehurst et al., 2024a), which we use for all Rocket transforms and dataset loading.

We then generate G independent groups of $k = 1{,}000$ random convolutional kernels (parameterized as in standard Rocket (Dempster et al., 2020)). Each group g transforms every instance into a $2k = 2{,}000$-dimensional feature vector, producing a feature matrix $\mathbf{F}_g \in \mathbb{R}^{n \times 2000}$ that aligns with TabPFN v2.5's recommended feature range.

Each feature matrix is passed to a TabPFN v2.5 classifier, which returns predicted probabilities $\mathbf{P}_g \in \mathbb{R}^{n_{\text{test}} \times C}$ via in-context learning (no gradient updates on the target data). The final prediction averages across groups:

$$\hat{y}_j = \arg\max_c \ \frac{1}{G} \sum_{g=1}^{G} P_g[j, c]. \tag{1}$$

This design is motivated by a simple constraint: TabPFN v2.5 is designed for up to 2,000 features, while standard Rocket uses 10,000 kernels (producing 20,000 features). Rather than truncating or compressing features, we distribute the kernels across G groups and ensemble the classifiers—a strategy that naturally provides diversity, since each group operates on a different random projection of the time series.

With $G = 10$ groups, RocketPFN uses the same total number of random kernels as default Rocket ($10 \times 1{,}000 = 10{,}000$), enabling a controlled comparison that isolates the classifier: ridge regression versus TabPFN. The entire pipeline is *training-free*: the convolutional kernels are randomly generated (not learned), and TabPFN classifies via its pre-trained weights. No parameters are fitted to the target dataset.

4 Experiments

4.1 Datasets and Evaluation Protocol

We evaluate RocketPFN (G=1 and G=10) on the UCR archive (Dau et al., 2019) (univariate) and the UEA archive (Bagnall et al., 2018) (multivariate), restricting to datasets within TabPFN's operational scope ($C \leq 10$ classes, $n_{\text{train}} \leq 50{,}000$), yielding 103 UCR and 23 UEA datasets (126 total). Of the 128 UCR and 30 UEA datasets in the archives, 25 UCR and 7 UEA are excluded, all due to the class limit (C ranging from 11 to 60); no dataset in the archive exceeds the training-size limit. Excluded datasets are listed in Appendix B. Each pairwise comparison is restricted to the intersection of datasets for which both methods have results: baseline accuracies for HC2 and its components are taken from Middlehurst et al. (2021), who evaluated on a standard benchmark of 112 UCR and 26 UEA datasets (variable-length and other incompatible datasets excluded), so comparisons against those methods use that intersection.

All methods follow the same evaluation protocol: 30 stratified resamples per dataset, where resample 0 is the original archive train/test split and resamples 1–29 are generated by stratified shuffling with seeds 1–29, preserving class proportions. We report mean accuracy over the 30 resamples. Pairwise comparisons use the

Wilcoxon signed-rank test (Wilcoxon, 1945) ($p < 0.05$); rankings across multiple methods are summarized with critical difference (CD) diagrams (Demšar, 2006). Encoder results (MOMENT, Mantis, MantisV2) are computed by us under this protocol; all other baseline results are taken from Middlehurst et al. (2021). All RocketPFN runs use the `aeon` toolkit (Middlehurst et al., 2024a) for dataset loading and feature extraction. Per-dataset results are in Appendix A.

4.2 Temporal Encoding is Necessary

We first ask whether TabPFN can classify time series *without* any temporal encoding, by treating each univariate series of length t as a flat vector of t raw values and passing it directly to TabPFN. Table 1 reports the comparison on the 90 UCR datasets that satisfy both $t \leq 2{,}000$, the feature range for which TabPFN is designed, and the HC2 bake-off benchmark for which published baseline numbers are available. The answer is clear: despite being state-of-the-art on tabular benchmarks, TabPFN flat is significantly outperformed by *every* established classifier on the UCR archive—including Rocket—and is statistically tied only with the two weakest baselines (Table 1).

Table 1: **TabPFN (flattened) vs. published baselines** on 90 UCR datasets (Wilcoxon signed-rank test, 30-resample protocol). TabPFN flat is significantly worse than all methods except TDE and STC.

Method	Mean Acc.	vs. TabPFN flat	p-value
HC2	0.900	+0.053	<0.001 ***
TS-CHIEF	0.884	+0.037	<0.001 ***
HC1	0.891	+0.044	<0.001 ***
InceptionTime	0.882	+0.035	0.005 **
ROCKET	0.882	+0.034	<0.001 ***
Arsenal	0.882	+0.035	<0.001 ***
DrCIF	0.883	+0.036	<0.001 ***
TDE	0.869	+0.022	0.138 n.s.
STC	0.871	+0.023	0.446 n.s.
TabPFN flat	0.847	—	—

The failure is not merely quantitative—it is structural. TabPFN is a powerful classifier, but without any inductive bias toward temporal structure it has no way to recover this information. Rocket features restore it by construction: convolutional max-pooling captures *whether* a pattern occurs regardless of *where* in the series, and PPV captures its prevalence—both of which are invariant to translation and robust to noise.

Figure 1 illustrates the failure mode with a concrete example. ShapeletSim ($t = 500$, $n_{\text{train}} = 20$) has two classes: both are Gaussian noise, but Class 1 contains a short triangular pulse at a *random* position. The pattern is immediately visible to a human, yet TabPFN flat scores ≈ 0.50—random chance. Encoding the same series with a single group of 1,000 Rocket kernels recovers near-perfect accuracy (≈ 1.00).

4.3 In-Context Learning Outperforms Ridge Regression on Random Convolutional Features

Before turning to the full state-of-the-art comparison, we isolate the contribution of the downstream classifier. At an *identical* kernel budget of 10,000 random kernels, RocketPFN (G=10, k=1,000) and Rocket differ only in the classifier applied to the resulting features: TabPFN in-context learning versus ridge regression with built-in cross-validated regularisation. On UCR, replacing ridge with TabPFN yields a significant accuracy gain of +0.018 (0.900 vs. 0.882, $p < 0.001$; Table 2, Rocket row), with RocketPFN winning 70 datasets, tying on 3 and losing on 19. The significant gain confirms that in-context learning extracts more discriminative information from the same random features than a regularised linear model can: TabPFN inherits Rocket's temporal inductive bias while replacing ridge's linear decision boundary with non-linear, context-dependent posterior predictions.

The effect is strong enough that RocketPFN remains competitive at a fraction of the kernel budget. With G=1 (only 1,000 kernels, a tenth of Rocket's default), RocketPFN already surpasses Rocket in mean

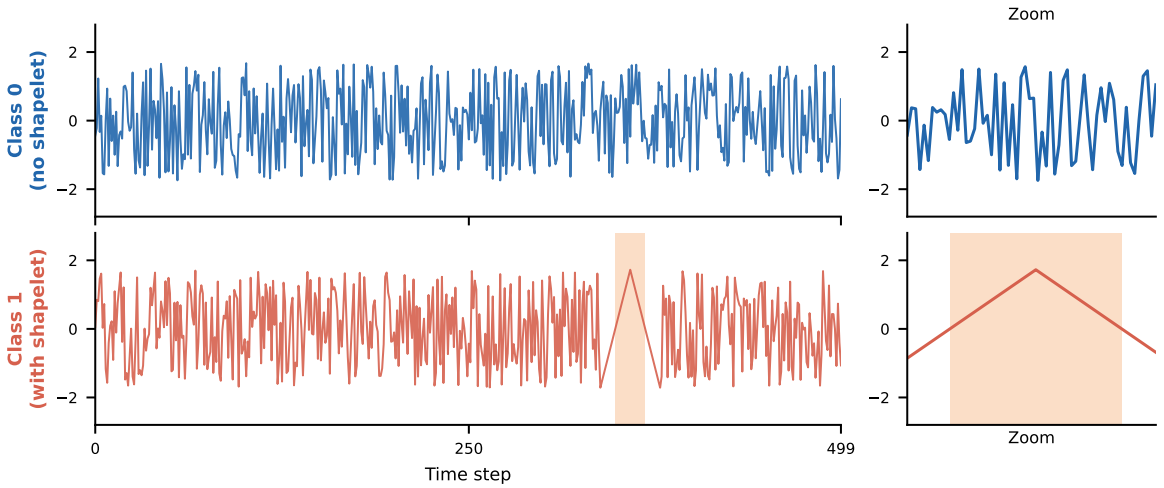


Figure 1: **ShapeletSim: trivial for humans, impossible for flattened features.** Class 0 is pure Gaussian noise; Class 1 contains a triangular pulse at a random position (highlighted in orange, enlarged right). TabPFN flat: ≈ 0.50 (chance). RocketPFN (G=1): ≈ 1.00 via translation-invariant pooling.

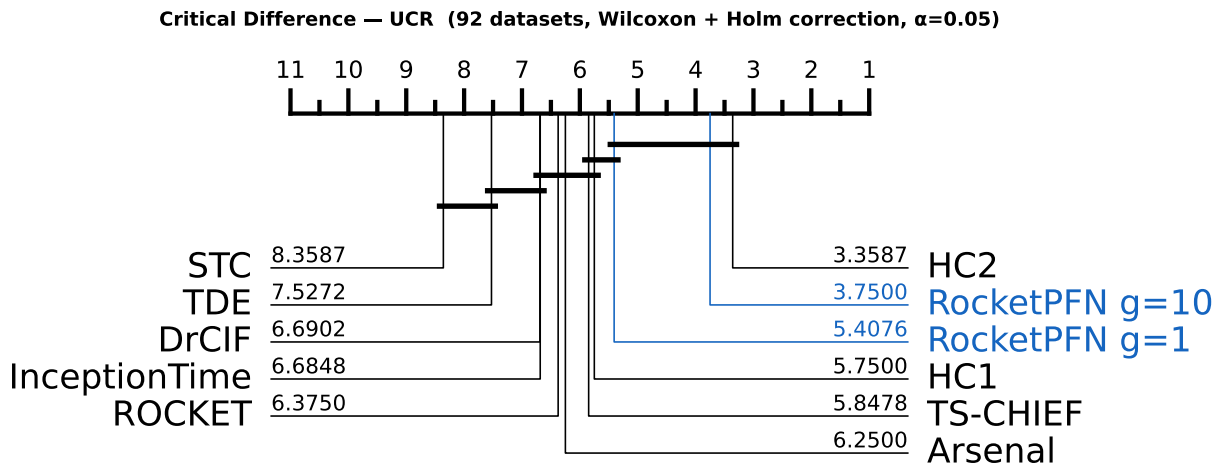
accuracy on UCR (0.894 vs. 0.882) and is not significantly different from it on UEA ($p = 0.70$; same 10× feature reduction).

4.4 RocketPFN Matches the State of the Art

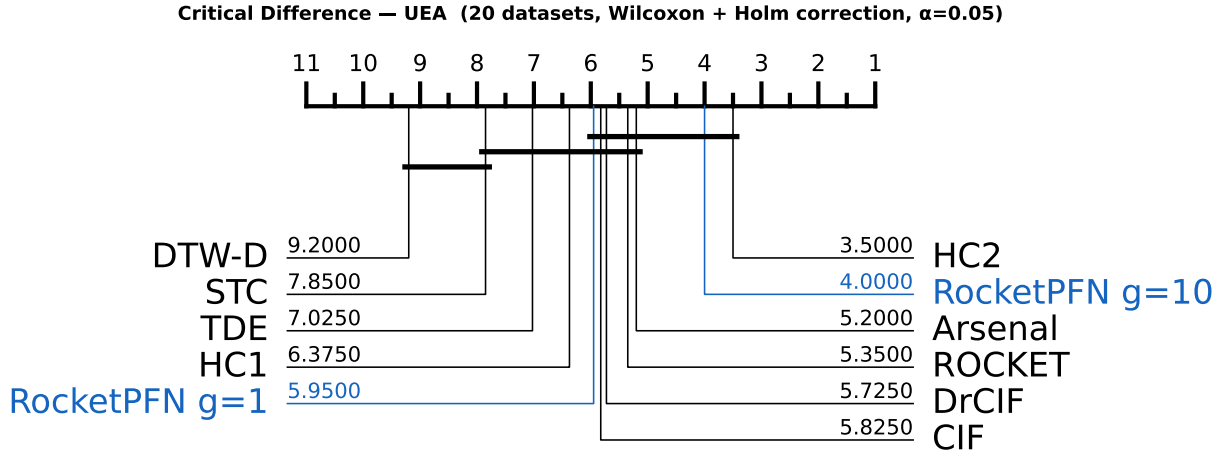
On UCR ($n = 92$), RocketPFN (G=10) achieves the same mean accuracy as HC2 (both 0.900, $p = 0.50$): a two-component, training-free pipeline reaches the same mean accuracy as a meta-ensemble of four specialised classifiers, each requiring hours of training. On UEA ($n = 20$), the benchmark is too small to draw strong statistical conclusions—with only 20 datasets, even sizable differences often fail to reach significance—but RocketPFN (G=10) achieves comparable mean accuracy to HC2 (both 0.747 at three decimal places, with HC2 ahead by less than 0.001, $p = 0.37$) and ranks near the top of all methods evaluated. Figure 2 shows the complete ranking; on UCR, RocketPFN and HC2 form a single clique and every other method falls strictly below.

RocketPFN (G=10) does not merely match HC2—it significantly outperforms every individual classifier in the HC2 ensemble: Arsenal, DrCIF, TDE, and STC, as well as InceptionTime, HC1, and TS-CHIEF ($p < 0.005$ on UCR for all; Table 2).

The result holds even at a tenth of the kernel budget. With G=1 (1,000 kernels, 2,000 features), RocketPFN already significantly outperforms all four HC2 components and Rocket on UCR ($p < 0.001$), and is not significantly different from HC2. On UEA, the 20-dataset benchmark limits statistical power throughout, and G=10 outperforms Rocket ($p = 0.033$); both configurations rank near the top across UEA datasets. On UCR, a single group of 1,000 random kernels already captures the temporal structure needed for competitive performance; G=10 closes the remaining gap with the full ensemble.



(a) UCR (92 datasets)



(b) UEA (20 datasets)

Figure 2: **Critical difference diagrams** (30-resample protocol, Wilcoxon + Holm correction, $\alpha = 0.05$). Thick bars connect methods with no significant difference. On UCR (92 datasets), RocketPFN (G=10) and HC2 are in the same clique; all other methods are ranked strictly below. On UEA (20 datasets), the small benchmark limits statistical power; RocketPFN (G=10) forms a single clique with HC2 and most other methods, consistent with competitive performance across multivariate datasets.

Table 2: **Pairwise comparisons on UCR** (Wilcoxon signed-rank test, 30-resample protocol) between RocketPFN and published baselines. Mean accuracy over 92 UCR datasets. W/T/L from the perspective of RocketPFN (G=10).

Method	n	**Mean Acc.**	**W/T/L vs. RocketPFN (G=10)**	**p-value**	**Sig.**
RocketPFN (G=10)	92	0.900	—	—	—
RocketPFN (G=1)	92	0.894	—	—	—
HC2	92	0.900	44/3/45	0.504	n.s.
HC1	92	0.891	57/2/33	0.004	**
TS-CHIEF	92	0.884	57/2/33	0.003	**
DrCIF	92	0.884	66/1/25	<0.001	***
InceptionTime	92	0.880	69/2/21	<0.001	***
Arsenal	92	0.882	69/3/20	<0.001	***
ROCKET	92	0.882	70/3/19	<0.001	***
STC	92	0.871	75/2/15	<0.001	***
TDE	92	0.870	65/2/25	<0.001	***

4.5 Random Features Outperform TSC Foundation Models

We now ask whether a TSC foundation model, used as a zero-shot feature extractor and paired with the same TabPFN classifier, can surpass random convolutional features. To rule out feature dimensionality as a confound from the start, we compare RocketPFN restricted to G=1, k=100 kernels (only 200 features—fewer than any of the three encoders, whose embedding dimension ranges from 256 (Mantis) to 1,024 (MOMENT)) against each encoder's full-dimensional representation. Under this stringent budget, RocketPFN still significantly outperforms all three encoders on 103 UCR datasets ($p < 0.001$ in each case; Table 3), winning 64–77 of 103 datasets.

Table 3: **RocketPFN vs. TSC foundation models** (Wilcoxon signed-rank test, 30-resample protocol). All TSC foundation models are used as zero-shot feature extractors with TabPFN as the downstream classifier. n = 103 UCR datasets. W/T/L and p-value are from the perspective of RocketPFN (G=1, k=100), which uses only 200 features—fewer than any of the three encoders—to rule out dimensionality as a confound.

Encoder	**Dim**	**Mean Acc.**	**W/T/L**	**p-value**	**Sig.**
RocketPFN (G=1, k=1 000)	2 000	0.887	—	—	—
RocketPFN (G=1, k=100)	200	0.874	—	—	—
RocketPFN (G=10, k=1 000)	2 000	0.894	—	—	—
Mantis + TabPFN	256	0.859	64/2/37	<0.001	***
MantisV2 + TabPFN	512	0.851	69/2/32	<0.001	***
MOMENT + TabPFN	1 024	0.845	77/1/25	<0.001	***

The finding is especially striking since Mantis and MOMENT were both pretrained on corpora that include the UCR training splits, eliminating pretraining-data mismatch as an explanation for the gap.

The gap is not an artifact of the downstream classifier, either: in the results reported by the respective papers, these encoders do not reach state-of-the-art accuracy on UCR under any of the classification heads they study (random forest, XGBoost, linear probe, fine-tuned MLP).

4.6 Computational Cost

RocketPFN requires *zero training on the target data*: Rocket applies a fixed random transform and TabPFN classifies via a single forward pass through a network pre-trained offline. There is no fitting step, no gradient computation, and no hyperparameter search at deployment time. On the 112 standard

benchmark datasets (92 UCR + 20 UEA), RocketPFN (G=10) completes in a median of **30.6 seconds per fold** (NVIDIA L40S GPU); with G=1 the median drops to **3.7 seconds per fold**.

4.7 Sensitivity Analysis

RocketPFN has three core design choices: the random convolutional feature extractor, the number of groups G, and the feature width per group. We evaluate three feature extractors: Rocket (Dempster et al., 2020), which produces two features per kernel (global max and PPV); MiniRocket (Dempster et al., 2021), a faster near-deterministic variant designed to scale the kernel count to 50,000; and MultiRocket (Tan et al., 2022), which targets higher accuracy with a richer pooling scheme. Both MiniRocket and MultiRocket were designed with a much larger kernel budget in mind; when the feature width is fixed at 2,000 per group (as required here by TabPFN's input limit), the three extractors operate in a similar regime, and we do not expect large differences. We vary the feature width per group in $\{672, 1344, 2016\}$ and $G \in \{1, 2, 5, 10, 25\}$, keeping e=8 fixed throughout, on the 92-dataset UCR benchmark.

Figure 3 shows mean accuracy as a function of G for each extractor, averaged over feature widths, with shaded bands indicating the standard deviation. All three extractors behave similarly: accuracy rises steeply from G=1 to G≈5 and plateaus thereafter, with differences between extractors below 0.006 across the plateau. The feature width has an equally muted effect across the range evaluated. No significant differences emerge between extractors. The same robustness extends to e: reducing TabPFN's internal ensemble from e=8 to e=1 costs less than 0.003 in mean accuracy once $G \geq 5$, as diversity across groups largely subsumes within-group ensembling.

The chosen settings—Rocket, $G \in \{1, 10\}$, k=1,000 (width 2,000)—were not the result of any optimisation or tuning. G=10 matches Rocket's default kernel budget for a clean comparison; G=1, k=100 matches or stays below the embedding dimension of TSC foundation models. Per-dataset selection of G or feature width could yield further gains.

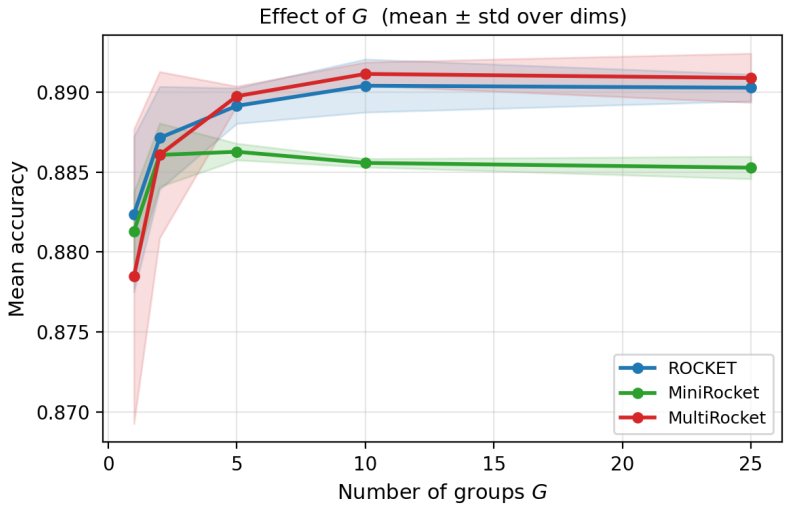


Figure 3: **Effect of G and feature extractor** (92 UCR datasets). Lines show mean accuracy as a function of G for Rocket, MiniRocket, and MultiRocket, averaged over feature widths $\{672, 1344, 2016\}$; shaded bands are ±1 standard deviation.

5 Conclusion

RocketPFN shows that a pipeline with no parameters fitted to the target data can match HC2 on UCR, by pairing random convolutional features with in-context tabular classification. The same pipeline outperforms

three recent TSC foundation models used zero-shot, even when restricted to fewer features than any of the learned encoders produce.

We read these results as evidence that the bar for zero-shot temporal representation learning is higher than current foundation models suggest. Future foundation models should justify why their pretraining cost and design complexity are worth it, at least by identifying concrete conditions under which their learned embeddings outperform RocketPFN, a training-free pipeline of random convolutional features classified in context.

Our findings complement concurrent work applying TabPFN to time series forecasting (Hoo et al., 2025), and together suggest a broader research direction: tabular pre-fitted networks may serve as a general backbone for classification problems well beyond structured tabular data, provided suitable feature extractors bridge the gap between the source modality and the tabular interface—a recipe that could extend to audio, images, graphs, and multimodal settings alike.

Limitations TabPFN v2.5 supports up to C=10 classes, which excludes 25 UCR and 7 UEA datasets (class counts ranging from 11 to 60; full list in Appendix B). This is a current, not a fundamental, limitation: any standard multi-class decomposition—one-vs-one, one-vs-rest, or error-correcting output codes (Dietterich and Bakiri, 1995)—reduces a C-class problem to a collection of $\leq$ 10-class subproblems compatible with TabPFN, and future versions of TabPFN may relax the bound directly. We chose not to introduce a decomposition strategy here because it would mix our zero-shot result with an engineering choice that is not central to the paper's claim. The intersection of the UEA bake-off benchmark in Middlehurst et al. (2021) with TabPFN-compatible datasets contains only 20 datasets, which is near the practical lower bound for the Wilcoxon signed-rank test: even sizable per-dataset differences frequently fail to reach $\alpha = 0.05$. We report the UEA comparison for completeness but treat it as inconclusive: RocketPFN and HC2 are tied in mean accuracy (both 0.747) and statistically indistinguishable ($p = 0.37$). RocketPFN is evaluated on GPU while baseline methods are reported on CPU. The reported timings should therefore not be read as a controlled hardware comparison, but rather as evidence of a structural gap: RocketPFN requires no per-dataset training phase, so its wall-clock time is bounded by inference regardless of scale.

References

Monica Arul and Ahsan Kareem. Applications of shapelet transform to time series classification of earthquake, wind and wave data. *Engineering Structures*, 228:111564, 2021.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

W. Art Chaovalitwongse, Ya-Ju Fan, and Rajesh C. Sachdeo. On the time series k-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 37(6):1005–1016, 2007.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Angus Dempster, François Petitjean, and Geoffrey I. Webb. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5): 1454–1495, 2020.

Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. MiniRocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 248–257, 2021.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

Vasilii Feofanov, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight calibrated foundation model for user-friendly time series classification, 2025. URL `https://arxiv.org/abs/2502.15637`.

Vasilii Feofanov, Songkang Wen, Jianfeng Zhang, Lujia Pan, and Ievgen Redko. MantisV2: Closing the zero-shot gap in time series classification with synthetic data and test-time strategies. *arXiv preprint arXiv:2602.17868*, 2026.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-3: Technical report, 2026. URL `https://arxiv.org/abs/2605.13986`.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: Extending TabPFN-v2 to time series forecasting. *arXiv preprint arXiv:2501.02945*, 2025.

Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

Yaguo Lei, Bin Yang, Xinwei Jiang, Feng Jia, Naipeng Li, and Asoke K. Nandi. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138:106587, 2020.

Jason Lines, Sarah Taylor, and Anthony Bagnall. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data*, 12(5):1–35, 2018.

Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony Bagnall. HIVE-COTE 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110:3211–3243, 2021.

Matthew Middlehurst, Ali Ismail-Fawaz, Antoine Guillaume, Christopher Holder, David Guijo-Rubio, Guzal Bulatova, Leonidas Tsaprounis, Lukasz Mentel, Martin Walter, Patrick Schäfer, and Anthony Bagnall. aeon: a Python toolkit for learning from time series. *arXiv preprint arXiv:2406.14231*, 2024a.

Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4): 1958–2031, 2024b. doi: 10.1007/s10618-024-01022-1.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Prior Labs Team. TabPFN-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

Ahmed Shifaz, Charlotte Pelletier, François Petitjean, and Geoffrey I. Webb. TS-CHIEF: A scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery*, 34(3): 742–775, 2020.

Chang Wei Tan, Angus Dempster, Christoph Bergmeir, and Geoffrey I. Webb. MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery*, 36(5):1623–1646, 2022.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Shifeng Xie, Vasilii Feofanov, Ambroise Odonnat, Lei Zan, Marius Alonso, Jianfeng Zhang, Themis Palpanas, Lujia Pan, Keli Zhang, and Ievgen Redko. CauKer: Classification time series foundation models can be pretrained on synthetic data. In *International Conference on Learning Representations (ICLR)*, 2026.

A Full Per-Dataset Results

Tables 4 and 5 report per-dataset mean accuracy (%) over 30 resamples. Blank entries indicate the method is not applicable to that dataset (e.g. Flat is blank when $t > 2{,}000$; HC2 is blank for datasets outside the standard benchmark of Middlehurst et al. (2021)).

Table 4: **UCR per-dataset results** (mean accuracy %, 30 resamples). Flat = TabPFN on raw flattened series ($t \leq 2{,}000$ only); Mantis, MantisV2, MOMENT each use the pretrained encoder with TabPFN as downstream classifier. **Bold** indicates the best result per dataset (ties included). HC2 results were obtained from the published benchmark of Middlehurst et al. (2021); all other results were produced by the authors.

Dataset	G=1	G=10	Flat	Mantis	MantisV2	MOMENT	HC2
ACSF1	86.0	**87.4**	77.6	70.2	72.1	70.0	83.3
AllGestureWiimoteX	75.8	**77.2**	59.8	66.8	60.4	66.3	
AllGestureWiimoteY	79.1	**80.3**	64.6	66.6	58.4	72.6	
AllGestureWiimoteZ	73.2	**75.0**	54.9	67.7	60.0	62.9	
ArrowHead	87.0	**88.8**	81.9	78.3	84.4	83.8	88.6
BME	99.7	99.7	98.4	99.2	81.5	96.1	**100.0**
Beef	76.3	77.0	**87.0**	60.1	63.2	74.3	79.7
BeetleFly	88.5	89.2	75.2	87.8	88.0	**94.7**	90.3
BirdChicken	92.2	92.2	78.3	87.8	89.7	87.2	**95.2**
CBF	99.9	**100.0**	94.3	99.9	99.7	98.4	99.8
Car	91.4	**92.2**	80.7	88.2	88.6	81.8	90.8
Chinatown	96.9	97.1	**97.6**	87.2	94.3	97.3	97.4
ChlorineConcentration	78.0	79.5	**96.6**	80.3	80.9	80.9	76.9
CinCECGTorso	86.9	88.5	91.3	74.7	75.4	73.1	**99.8**
Coffee	**100.0**	**100.0**	**100.0**	99.5	**100.0**	99.2	99.9
Computers	87.8	**88.8**	70.7	77.6	80.1	70.8	85.9
DiatomSizeReduction	95.4	95.6	95.3	90.5	90.8	**95.8**	92.2
DistalPhalanxOutlineAgeGroup	79.2	80.2	78.8	78.6	79.4	80.2	**82.1**
DistalPhalanxOutlineCorrect	83.7	83.9	82.9	82.5	**84.7**	83.1	82.8
DistalPhalanxTW	70.6	71.7	70.5	**72.1**	71.1	70.5	70.2
DodgerLoopDay	58.7	**61.5**	61.3	54.2	49.1	47.2	
DodgerLoopGame	83.4	**84.1**	75.0	73.1	60.0	76.9	
DodgerLoopWeekend	98.3	98.4	**98.6**	96.9	96.0	96.6	
ECG200	90.4	**90.8**	86.0	83.2	86.2	88.5	88.9
ECG5000	94.7	94.7	94.3	94.1	94.3	94.6	**94.8**

Continued on next page

Table 4 – continued

Dataset	G=1	G=10	Flat	Mantis	MantisV2	MOMENT	HC2
ECGFiveDays	99.1	99.2	92.2	92.8	93.7	98.0	**99.7**
Earthquakes	74.5	74.8	**74.9**	74.8	74.8	73.6	74.8
ElectricDevices	91.1	**91.6**	85.5	89.3	88.4	81.7	89.8
EthanolLevel	84.8	86.6	**94.2**	60.8	61.4	80.7	83.6
FaceFour	91.5	92.2	90.4	95.8	93.6	87.4	**96.7**
Fish	97.4	97.6	91.7	95.0	94.9	94.5	**98.2**
FordA	94.8	95.1	92.6	91.2	92.9	92.9	**95.4**
FordB	92.7	**93.1**	89.8	89.7	90.4	91.1	93.0
FreezerRegularTrain	99.9	99.9	99.8	99.4	98.9	99.2	**100.0**
FreezerSmallTrain	99.1	99.3	98.3	95.2	93.7	87.9	**99.8**
GesturePebbleZ1	97.2	**97.5**	92.3	96.1	95.3	93.8	
GesturePebbleZ2	97.4	**98.0**	92.5	96.5	96.0	94.3	
GunPoint	99.4	99.5	96.0	99.2	98.3	97.9	**99.9**
GunPointAgeSpan	99.2	99.2	98.4	99.5	**99.6**	97.1	**99.6**
GunPointMaleVersusFemale	99.9	99.9	99.9	99.8	99.7	99.5	**100.0**
GunPointOldVersusYoung	99.1	99.1	99.9	99.9	**100.0**	97.9	**100.0**
Ham	84.7	85.7	84.6	77.1	77.2	79.3	**85.9**
HandOutlines	**95.4**	95.2		93.3	93.7	91.7	93.5
Haptics	54.2	**55.7**	48.2	52.2	51.6	48.4	54.7
Herring	62.2	**63.5**	61.0	63.0	59.8	60.4	62.6
HouseTwenty	96.1	96.2	86.1	95.7	93.6	90.3	**98.6**
InlineSkate	55.7	**58.3**	37.3	42.3	44.2	39.4	54.6
InsectEPGRegularTrain	98.9	98.9	**100.0**	**100.0**	**100.0**	93.7	**100.0**
InsectEPGSmallTrain	91.8	92.2	**100.0**	**100.0**	**100.0**	82.9	99.9
ItalyPowerDemand	95.9	96.2	**96.4**	91.4	91.2	95.5	96.2
LargeKitchenAppliances	94.8	**95.1**	73.6	88.7	82.9	85.9	93.5
Lightning2	81.4	**83.6**	72.3	81.5	75.7	80.1	78.4
Lightning7	80.4	**82.0**	67.5	77.0	73.7	74.7	79.8
Mallat	97.3	97.4	96.8	96.6	95.2	96.1	**97.9**
Meat	98.8	98.8	99.7	99.1	99.6	**99.8**	99.1
MedicalImages	83.6	**84.6**	81.2	77.3	78.7	80.4	81.0
MelbournePedestrian	90.8	91.3	**97.9**	94.4	95.8	89.3	
MiddlePhalanxOutlineAgeGroup	62.9	65.4	70.5	70.5	**71.8**	69.9	**71.8**
MiddlePhalanxOutlineCorrect	84.7	**85.2**	85.1	82.8	83.9	85.0	83.6
MiddlePhalanxTW	56.3	58.2	58.4	57.5	58.0	58.1	**59.2**
MixedShapesRegularTrain	97.0	97.2	94.3	96.5	95.7	92.8	**97.3**
MixedShapesSmallTrain	93.8	94.3	84.2	92.3	91.1	84.2	**95.2**
MoteStrain	90.3	91.1	86.7	88.2	90.1	87.9	**93.3**
OSULeaf	95.8	96.4	67.1	93.7	92.7	84.0	**96.6**
OliveOil	89.4	90.0	90.9	90.0	**91.7**	88.0	90.3
PhalangesOutlinesCorrect	86.4	**86.6**	**86.6**	84.0	86.1	86.0	84.5
PickupGestureWiimoteZ	76.3	78.7	69.1	**80.4**	78.7	70.5	
Plane	**100.0**	**100.0**	98.9	**100.0**	**100.0**	99.6	**100.0**
PowerCons	95.8	96.2	**100.0**	93.1	92.6	90.7	96.9
ProximalPhalanxOutlineAgeGroup	83.5	84.3	85.1	84.3	84.0	84.6	**85.4**
ProximalPhalanxOutlineCorrect	90.6	**91.0**	90.6	89.1	90.1	89.8	89.6
ProximalPhalanxTW	80.9	81.5	**82.3**	79.9	80.9	80.7	81.1
RefrigerationDevices	80.6	**81.3**	54.5	69.5	71.6	66.9	76.7
Rock	74.8	76.5		77.6	77.3	66.8	**88.9**
ScreenType	71.1	**72.6**	48.9	58.4	58.0	53.3	70.7
SemgHandGenderCh2	95.9	**96.5**	96.0	93.0	87.2	87.9	96.1
SemgHandMovementCh2	86.0	**87.9**	82.7	80.2	72.3	62.2	86.2
SemgHandSubjectCh2	95.0	**95.8**	95.3	91.1	83.0	85.4	93.9
ShakeGestureWiimoteZ	88.1	89.4	76.7	**90.1**	89.3	88.7	
ShapeletSim	99.9	**100.0**	49.6	92.5	80.0	92.4	**100.0**
SmallKitchenAppliances	84.6	**85.1**	75.4	82.7	81.2	70.4	84.1
SmoothSubspace	96.5	97.5	**100.0**	96.4	97.4	81.4	98.3
SonyAIBORobotSurface1	95.5	**95.9**	86.4	89.5	89.3	90.2	95.2
SonyAIBORobotSurface2	92.5	93.3	86.8	84.7	88.7	89.0	**94.7**
StarLightCurves	98.2	**98.3**	97.7	98.0	98.1	96.9	98.2
Strawberry	98.2	98.3	**98.5**	97.4	97.9	98.0	97.9
Symbols	97.0	**97.1**	85.9	95.7	93.9	95.4	**97.1**
SyntheticControl	99.6	**99.8**	98.8	99.4	99.1	99.1	**99.8**
ToeSegmentation1	93.4	93.6	56.7	**95.7**	94.3	91.5	95.5
ToeSegmentation2	92.8	93.5	66.6	94.9	93.3	92.3	**96.0**
Trace	**100.0**	**100.0**	95.4	**100.0**	**100.0**	**100.0**	**100.0**
TwoLeadECG	99.6	99.7	94.5	99.3	99.2	91.3	**99.9**
TwoPatterns	**100.0**	**100.0**	99.8	98.7	99.4	99.6	**100.0**
UMD	98.6	98.6	**99.3**	98.6	98.7	96.4	98.6
UWaveGestureLibraryAll	97.4	**97.7**	97.1	92.3	89.9	96.4	97.5
UWaveGestureLibraryX	87.0	**87.5**	81.6	83.8	82.1	83.9	86.0
UWaveGestureLibraryY	80.4	**81.1**	73.8	76.2	74.6	76.6	78.9
UWaveGestureLibraryZ	81.9	**82.5**	75.9	79.4	78.1	77.7	80.3
Wafer	99.9	99.9	99.6	99.8	99.7	99.7	**100.0**
Wine	93.5	94.1	94.2	94.4	93.7	**95.0**	92.5
Worms	**77.4**	77.1	59.6	71.1	73.8	67.4	75.8
WormsTwoClass	78.1	78.2	60.8	74.5	76.5	74.9	**80.5**

Continued on next page

Table 4 – continued

Dataset	G=1	G=10	Flat	Mantis	MantisV2	MOMENT	HC2
Yoga	91.0	91.7	89.0	86.8	88.1	88.7	**93.0**

Table 5: **UEA per-dataset results** (mean accuracy %, 30 resamples). HC2 is blank for datasets outside the standard benchmark of Middlehurst et al. (2021). **Bold** indicates the best result per dataset (ties included). HC2 results were obtained from the published benchmark of Middlehurst et al. (2021); all other results were produced by the authors.

Dataset	G=1	G=10	HC2
AtrialFibrillation	25.6	27.8	**28.2**
BasicMotions	98.3	98.6	**98.9**
DuckDuckGeese	43.7	48.8	**49.9**
ERing	96.9	97.4	**98.5**
EigenWorms	91.3	**91.9**	89.4
Epilepsy	99.3	99.6	**99.8**
EthanolConcentration	68.8	73.3	**79.1**
FaceDetection	64.2	68.9	**71.3**
FingerMovements	56.2	**58.4**	55.2
HandMovementDirection	38.7	**46.4**	39.8
Heartbeat	71.3	72.4	**72.9**
InsectWingbeat	28.7	**38.8**	
JapaneseVowels	83.1	**84.9**	
MotorImagery	51.4	51.7	**53.2**
NATOPS	88.2	88.9	**89.2**
PEMS-SF	92.9	96.6	**99.8**
PenDigits	**99.7**	**99.7**	99.6
RacketSports	91.3	92.6	**93.0**
SelfRegulationSCP1	86.7	**87.9**	**87.9**
SelfRegulationSCP2	51.6	**53.4**	50.5
SpokenArabicDigits	99.5	**99.6**	
StandWalkJump	**44.9**	43.8	43.8
UWaveGestureLibrary	94.7	**95.3**	94.9

B Excluded Datasets

TabPFN v2.5 supports up to $C = 10$ classes and $n_{\text{train}} = 50{,}000$ training samples. Of the 128 UCR datasets, 25 are excluded solely due to the class constraint (C ranging from 11 to 60): Adiac (C=37), CricketX/Y/Z (C=12), Crop (C=24), EOGHorizontalSignal (C=12), EOGVerticalSignal (C=12), FaceAll (C=14), FacesUCR (C=14), FiftyWords (C=50), Fungi (C=18), GestureMidAirD1/D2/D3 (C=26), InsectWingbeatSound (C=11), NonInvasiveFetalECGThorax1/2 (C=42), Phoneme (C=39), PigAirwayPressure/ArtPressure/CVP (C=52), PLAID (C=11), ShapesAll (C=60), SwedishLeaf (C=15), and WordSynonyms (C=25). Of the 30 UEA datasets, 7 are likewise excluded: ArticularyWordRecognition (C=25), CharacterTrajectories (C=20), Cricket (C=12), Handwriting (C=26), Libras (C=15), LSST (C=14), and PhonemeSpectra (C=39). No dataset in either archive exceeds the training-size limit.

The class constraint could be overcome with standard multi-class decomposition strategies such as one-vs-one, one-vs-rest, or error-correcting output codes (Dietterich and Bakiri, 1995), each of which reduces a C-class problem to a collection of binary (or smaller-class) subproblems compatible with TabPFN. The training-size limit—not currently binding—could similarly be addressed through ensemble subsampling, where each of the G TabPFN instances operates on a random subsample of the training set, naturally extending the approach to larger datasets as tabular foundation models continue to scale.

C RocketPFN with TabPFN v3

Setup. TabPFN v3 (Grinsztajn et al., 2026) was released after the experiments in the main text were completed. We re-ran the two TabPFN-dependent parts of our study, TabPFN applied to flattened series and the full RocketPFN pipeline, under TabPFN v3, keeping the feature extractor, hyperparameters, and 30-resample protocol of Section 4.1 unchanged, and we compare on the same 92 UCR datasets used in the main text. The larger class limit of TabPFN v3 additionally makes 26 datasets usable that TabPFN v2.5 cannot process ($C > 10$; see Appendix B), which we do not add here.

Temporal encoding is still necessary. TabPFN v3 applied to raw flattened series reaches a mean accuracy of only 0.844, far below the encoded pipeline, and still scores at chance (49.6) on ShapeletSim despite v3's revised training prior. The Rocket front-end remains essential.

RocketPFN with TabPFN v3 vs. v2.5. Figure 4 compares the two TabPFN versions inside RocketPFN at three settings: a single group of 100 kernels (200 features), a single group of 1,000 kernels (2,000 features), and the default of ten groups of 1,000 kernels each. Each bar is the mean accuracy over the 92 UCR datasets. The outcome splits by feature budget: RocketPFN with TabPFN v3 beats v2.5 at the 200-feature setting (0.882 vs. 0.881), but loses at the 2,000-feature settings, both with one group (0.893 vs. 0.894) and with ten (0.895 vs. 0.900). Raising TabPFN v3's internal ensemble to e=16 does not reverse this: v3 stays ahead at 200 features and behind at 2,000. We use TabPFN v2.5 throughout the main text.

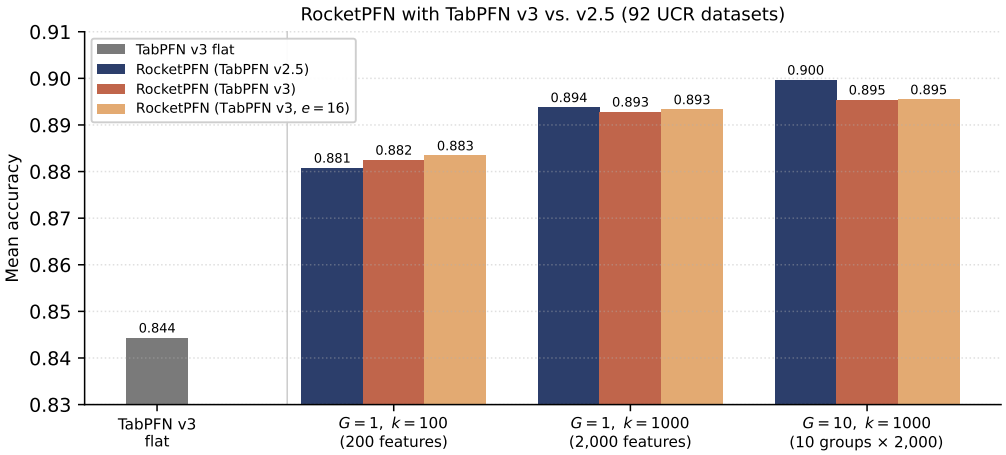


Figure 4: **RocketPFN with TabPFN v3 vs. v2.5** (mean accuracy over 92 UCR datasets, 30-resample protocol). The leftmost bar is TabPFN v3 applied to flattened series, which stays far below. The remaining groups show RocketPFN at three settings, comparing TabPFN v2.5 with TabPFN v3 under its automatic internal ensemble and under a fixed e=16. TabPFN v3 is ahead of v2.5 at the 200-feature setting and behind it at the 2,000-feature settings.