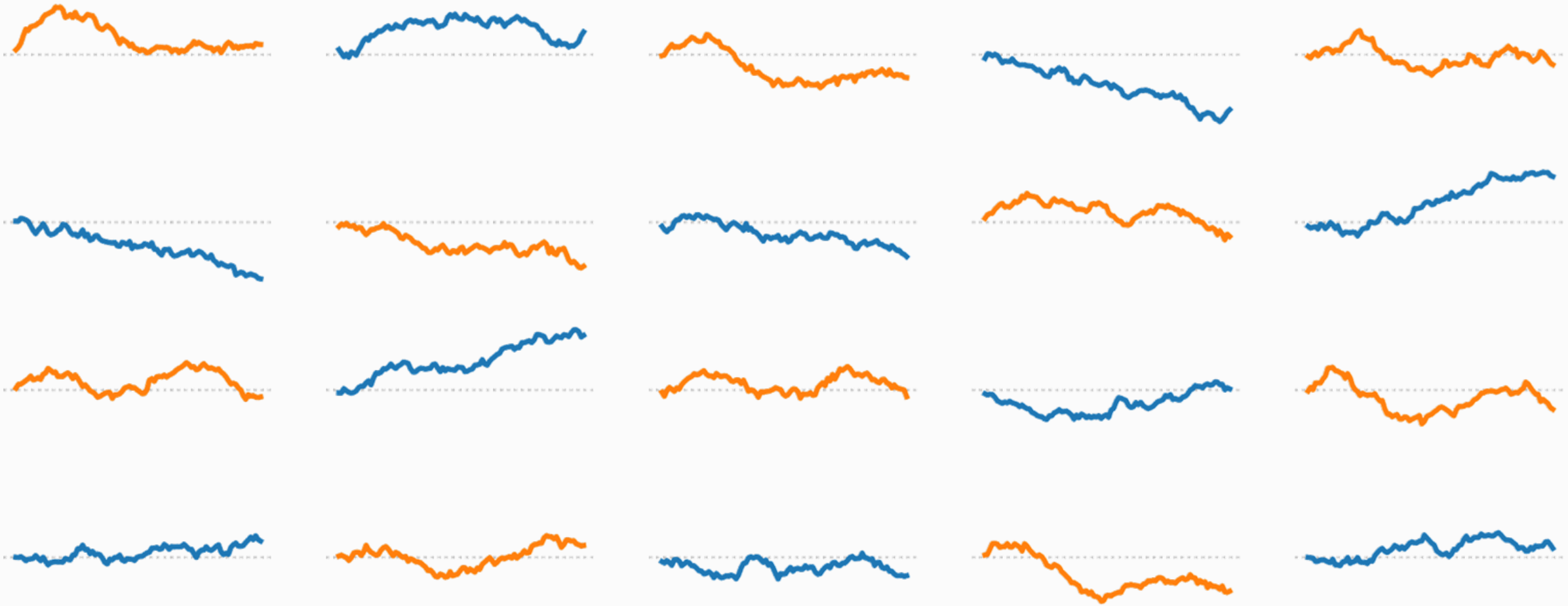
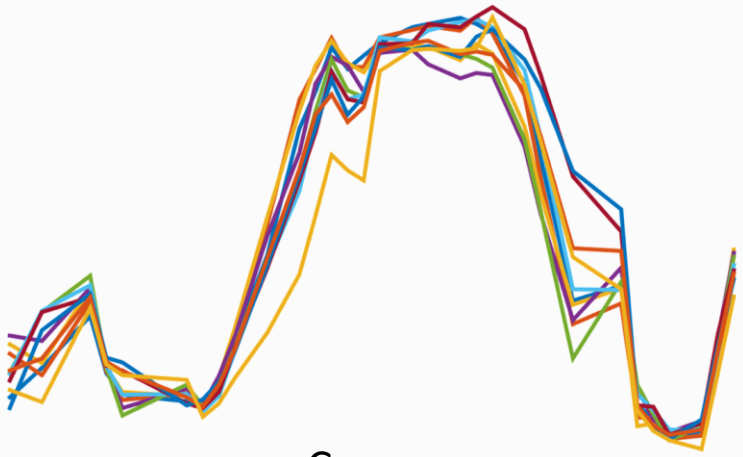
Convolutional Kernels for Efficient and Effective Time Series Analytics

Geoff Webb
Monash University
http://i.giwebb.com
Work with A. Dempster, C. Bergmeir, F. Petitjean, D. Schmidt, C. W. Tan

Time series classification

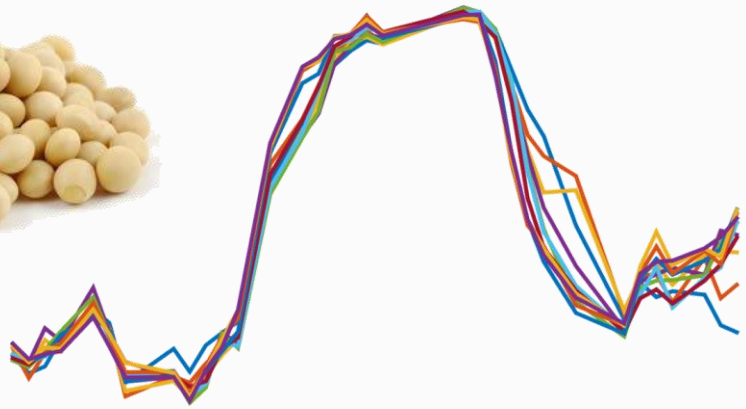


Land use mapping from earth observations

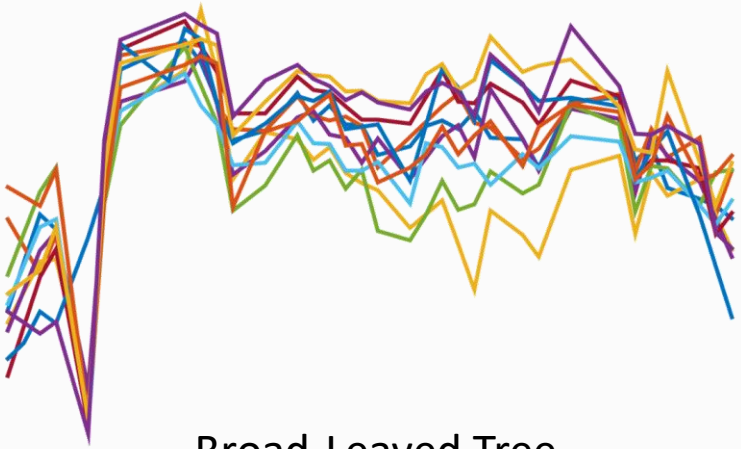


Corn

Wheat



Soybean

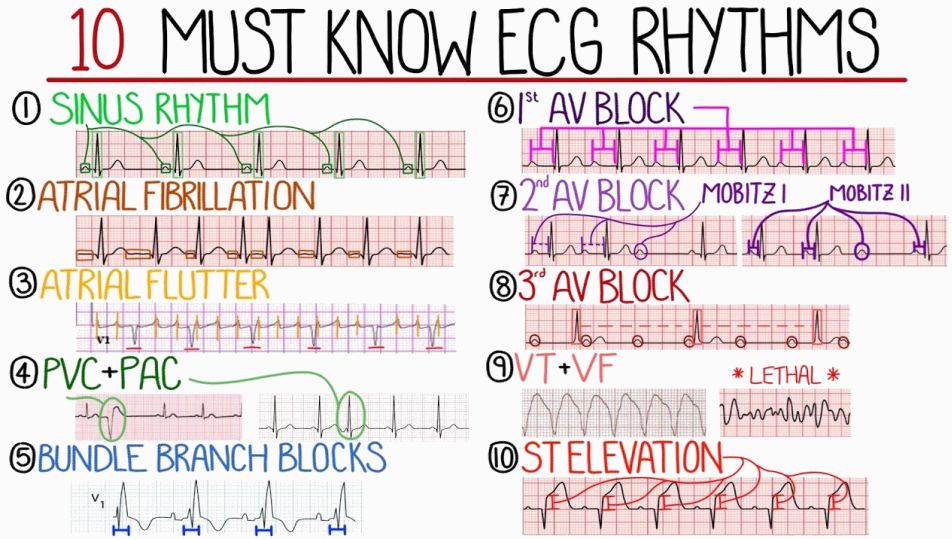


Broad-Leaved Tree

Many aspects of a series may be relevant

- From ONE ECG:
  - Frequency
    - Racing vs normal pulse
  - Variance in frequency
    - Irregular vs normal heartbeat
  - Amplitude
    - Strong vs weak pulse
  - Local pattern
    - Physiology of the heart and arteries
  - Global pattern
    - Declining or improving heartbeat

https://www.heart.org/en/health-topics/heart-attack/diagnosing-a-heart-attack/electrocardiogram-ecg-or-ekg

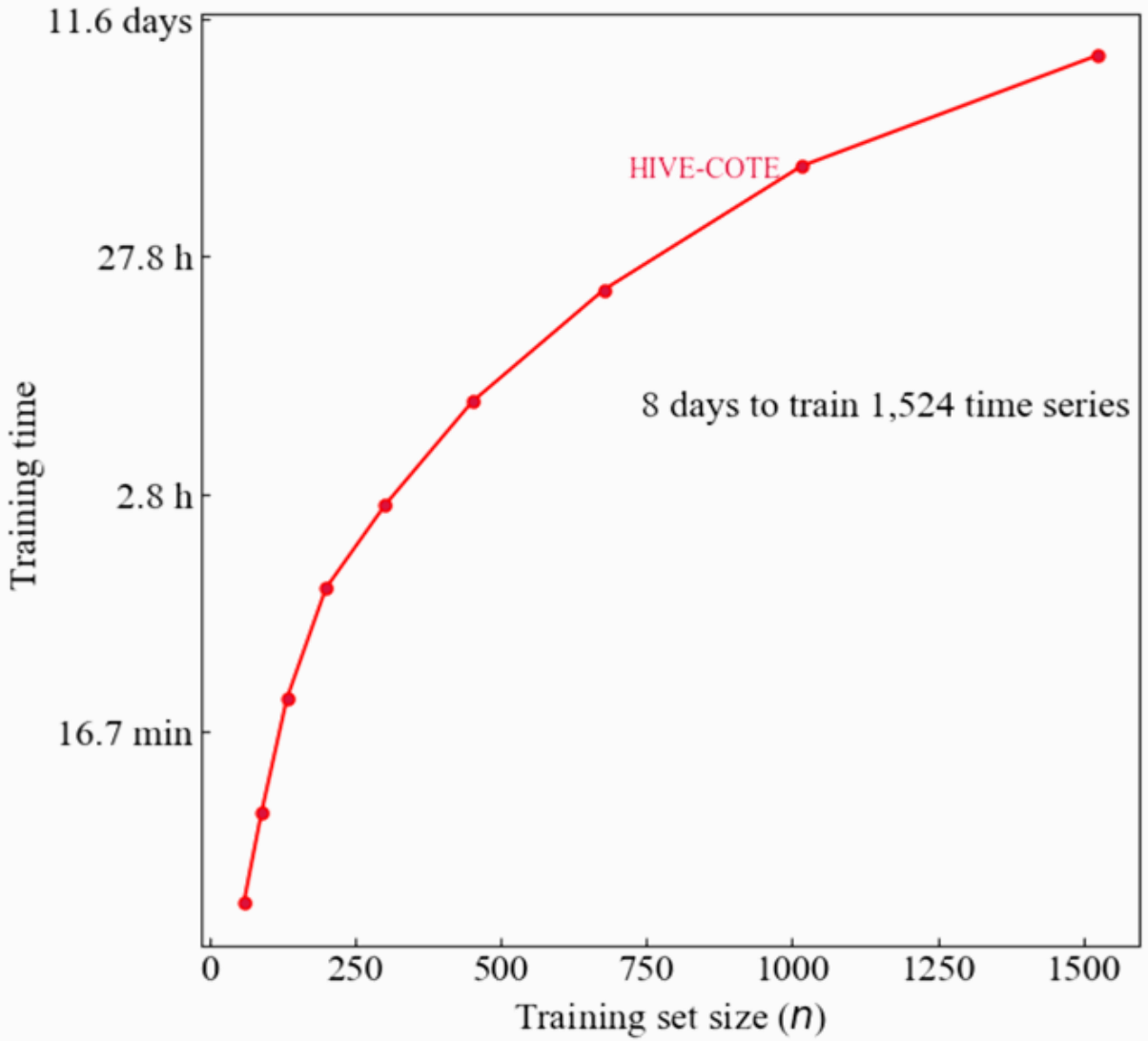


https://www.youtube.com/watch?v=vq3ba4BhddM

Many specialized techniques

- Nearest neighbor with specialized similarity measures
- Shapelets
- Dictionary
- Interval statistics
- Deep learning
- Ensembles

However, circa 2019 the most accurate classifiers did not scale

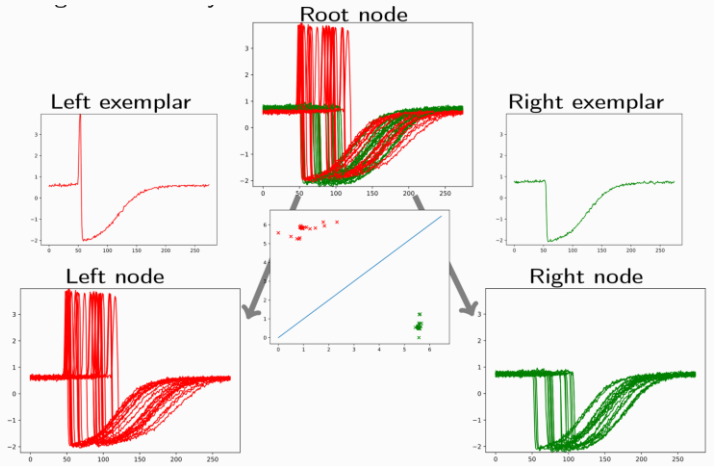


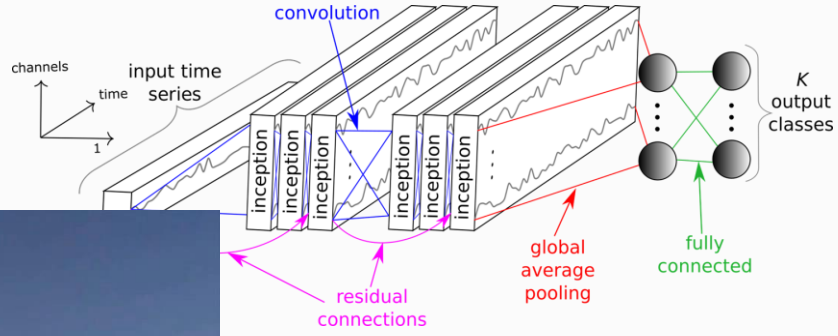
200+ years to learn from 1 million series!

Our accurate and scalable TSCs

- Tree-based: *Proximity Forest* and *TS-CHIEF*
- Deep Learning: *InceptionTime*
- A revolution:

ROCKET





ROCKET

2020 – Data Mining and Knowledge Discovery

Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-020-00701-z

ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels

Angus Dempster[1] · François Petitjean[1] · Geoffrey I. Webb[1]

Received: 28 October 2019 / Accepted: 18 June 2020
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2020

Abstract
Most methods for time series classification that attain state-of-the-art accuracy have high computational complexity, requiring significant training time even for smaller datasets, and are intractable for larger datasets. Additionally, many existing methods focus on a single type of feature such as shape or frequency. Building on the recent success of convolutional neural networks for time series classification, we show that simple linear classifiers using random convolutional kernels achieve state-of-the-art accuracy with a fraction of the computational expense of existing methods. Using this method, it is possible to train and test a classifier on all 85 'bake off' datasets in the UCR archive in < 2 h, and it is possible to train a classifier on a large dataset of more than one million time series in approximately 1 h.

Keywords Scalable · Time series classification · Random · Convolution

1 Introduction

Most methods for time series classification that attain state-of-the-art accuracy have high computational complexity, requiring significant training time even for smaller datasets, and simply do not scale to large datasets. This has motivated the development of more scalable methods such as Proximity Forest (Lucas et al. 2019), TS-CHIEF

Responsible editor: Aristides Gionis, Carlotta Domeniconi, Eyke Hüllermeier, Ira Assent.

Angus Dempster
angus.dempster1@monash.edu
François Petitjean
francois.petitjean@monash.edu
Geoffrey I. Webb
geoff.webb@monash.edu

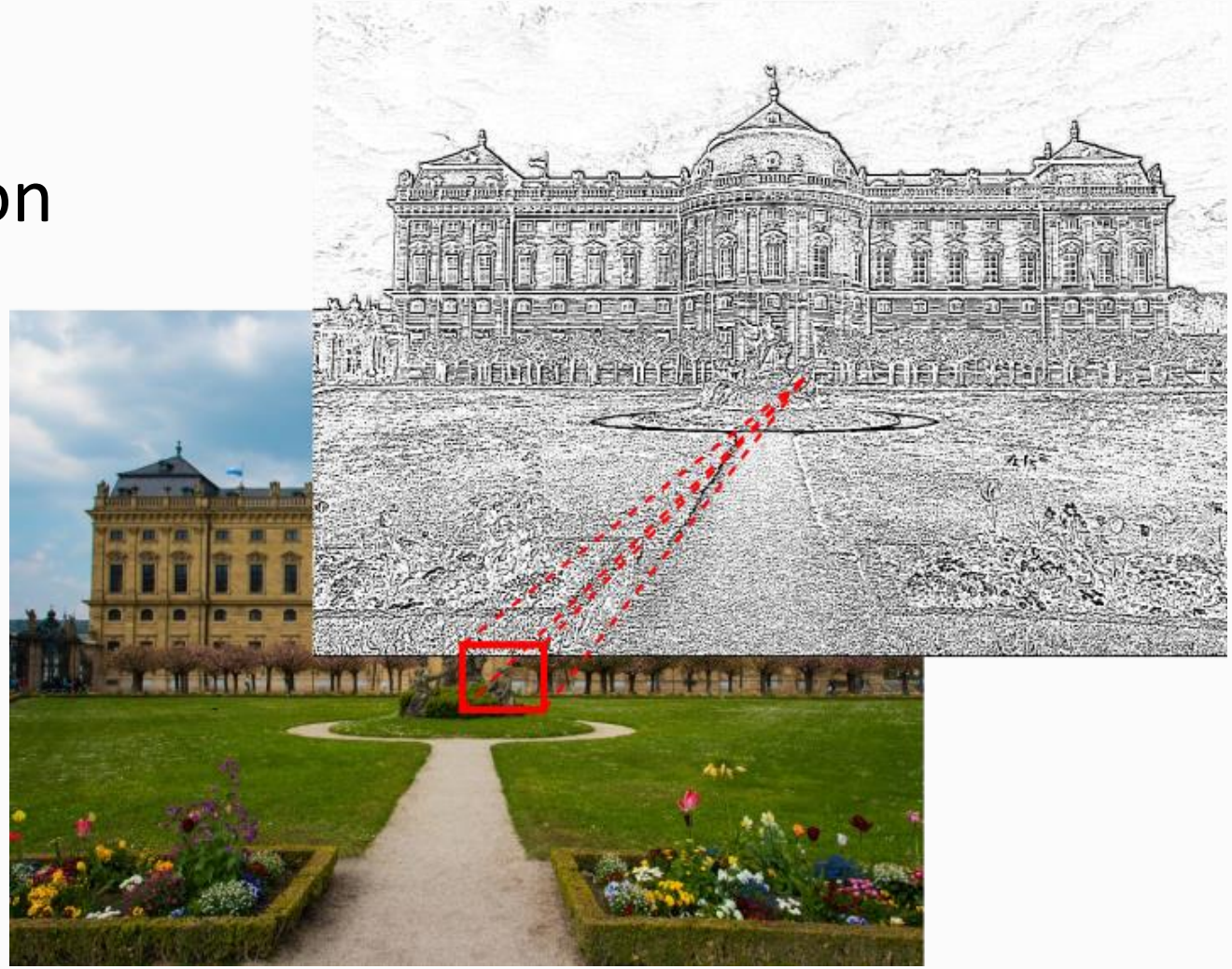
[1] Faculty of Information Technology, Monash University, Melbourne, Australia

Published online: 13 July 2020 Springer

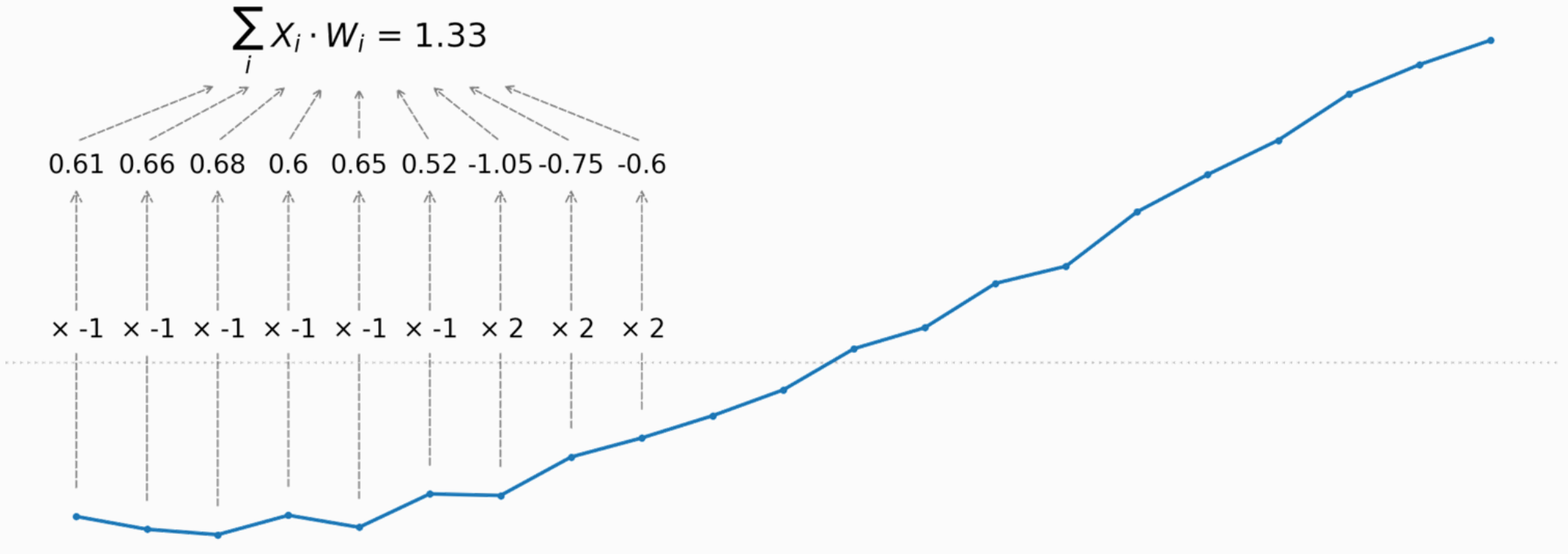
A. Dempster, F. Petitjean, G. Webb (2020). ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*. **34**(5): 1454 - 1495.
A. Dempster, D. Schmidt, G.Webb (2021) MINIROCKET: A Very Fast (Almost) Deterministic Transform for Time Series Classification. *Proceedings of the 27th SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 248-257, 2021.

Convolutions on images

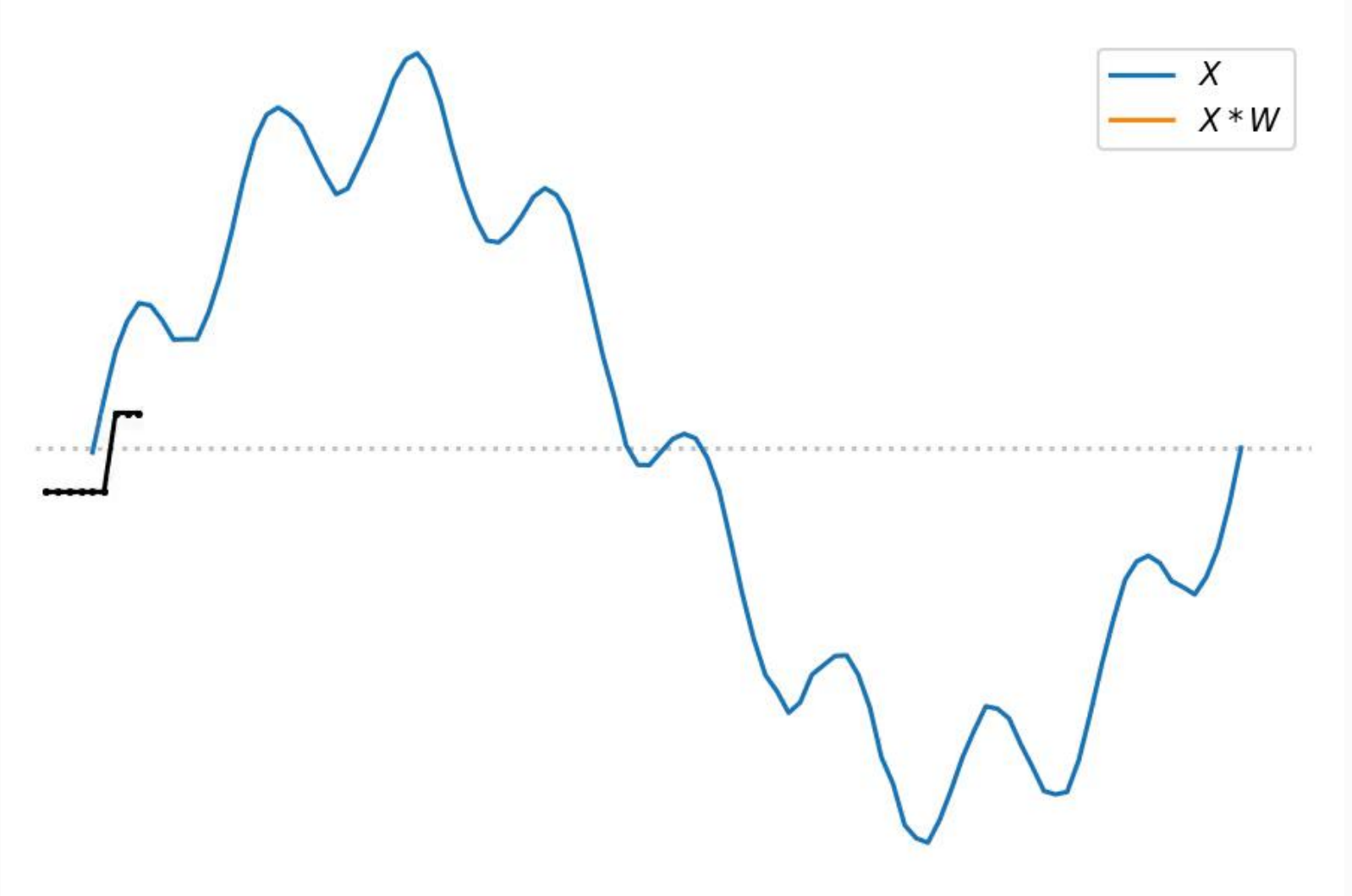
The result of applying an edge detection convolution on an image



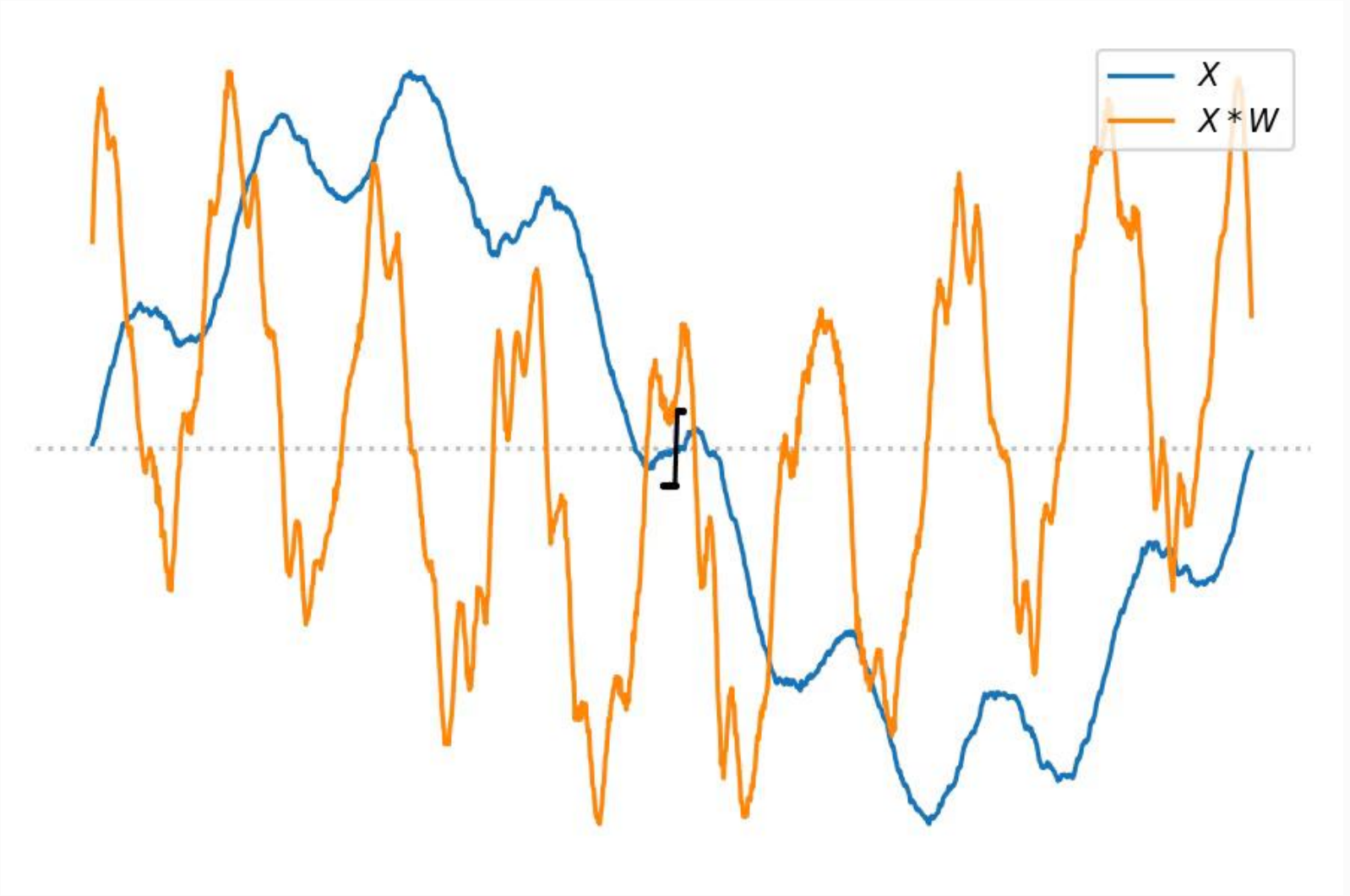
Time Series Convolutions



Time Series Convolutions

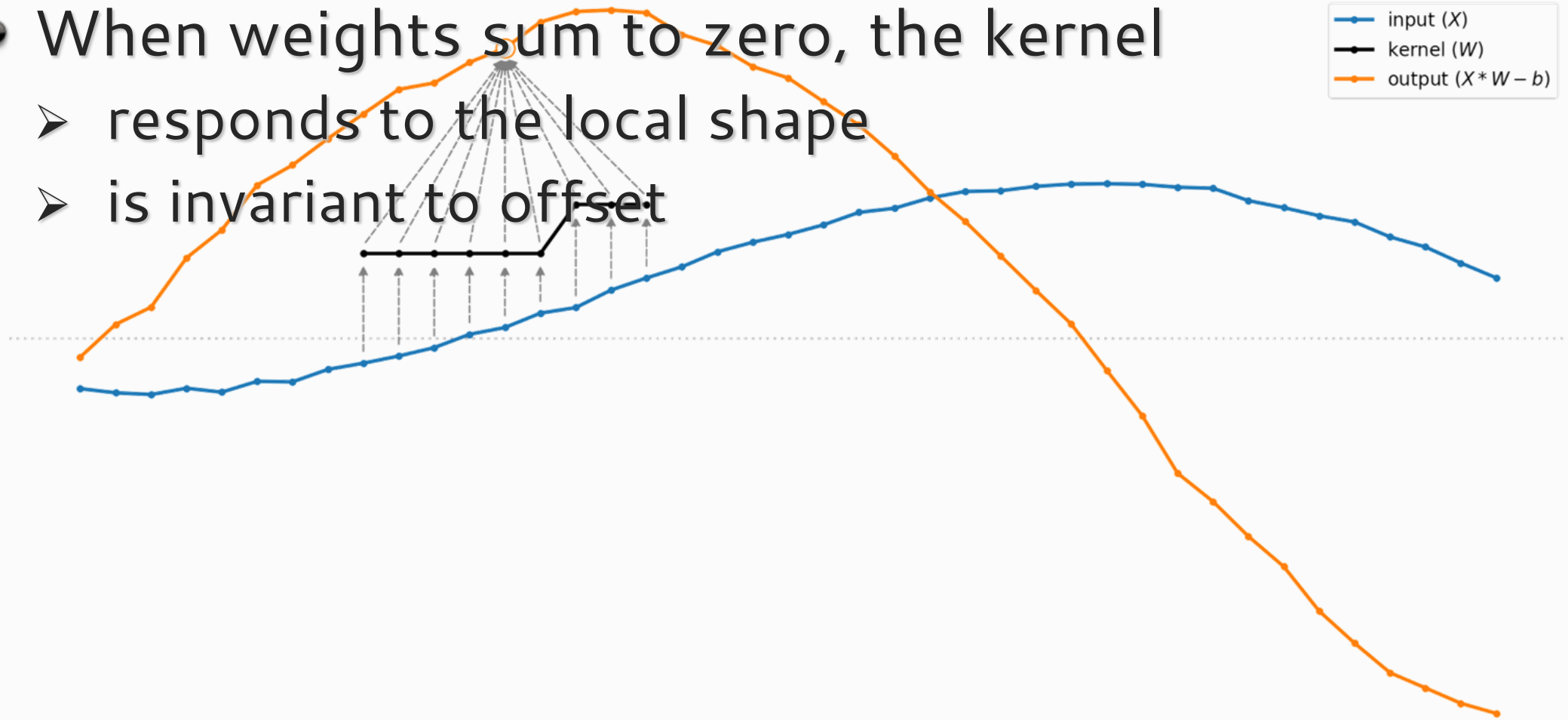


Dilation

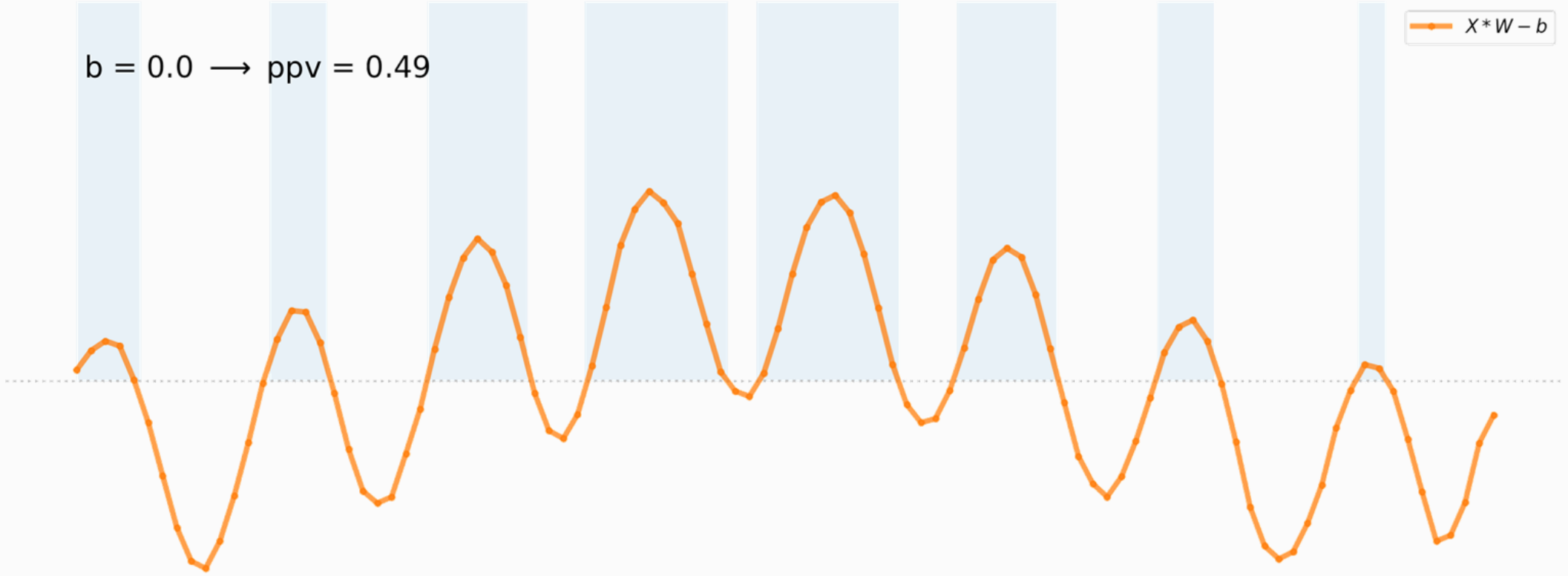


Sum to zero

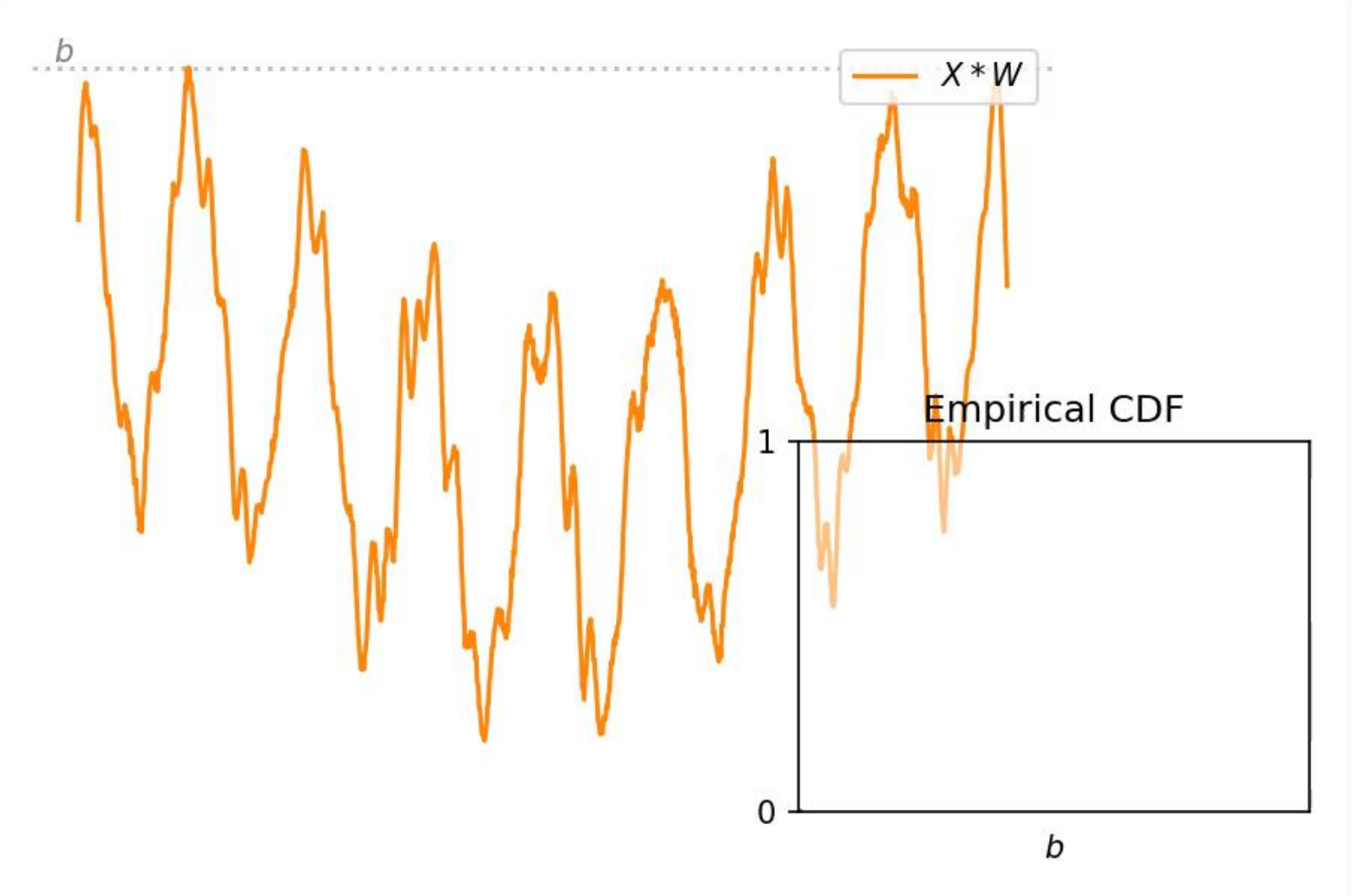
- When weights sum to zero, the kernel
  - responds to the local shape
  - is invariant to offset



Proportion of Positive Values (PPV)



PPV and Bias



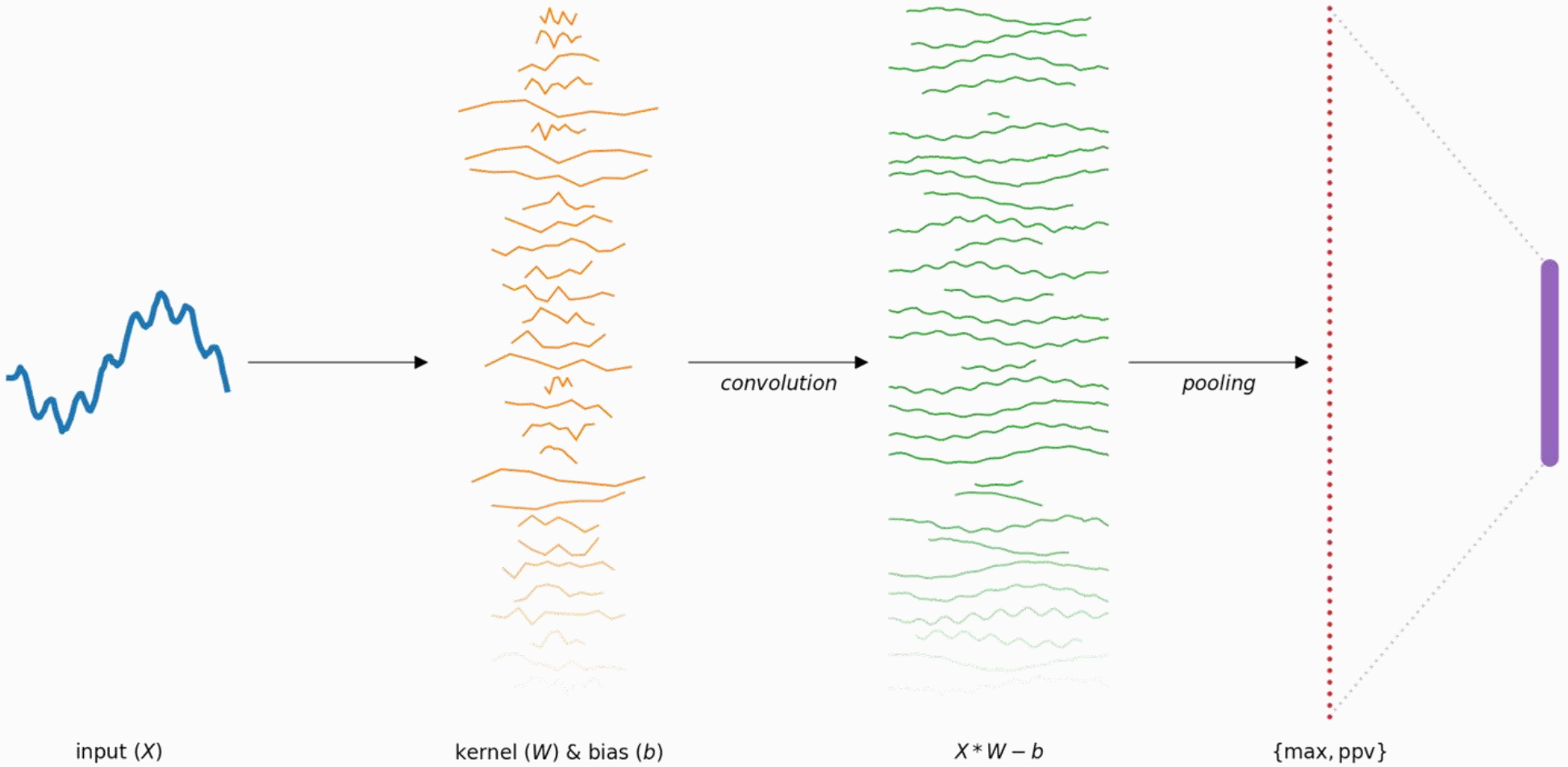
Exploit power of convolutional filters

- Convolutional filters can capture many different types of feature of a time series
  - Frequency, Amplitude, Local pattern, Global pattern
- Generate a large number – 10,000
- Learn simple linear classifier
  - Logistic regression when number of examples is large
  - Ridge regression when number of examples is small
    - Stronger regularisation
    - Faster for small sample size

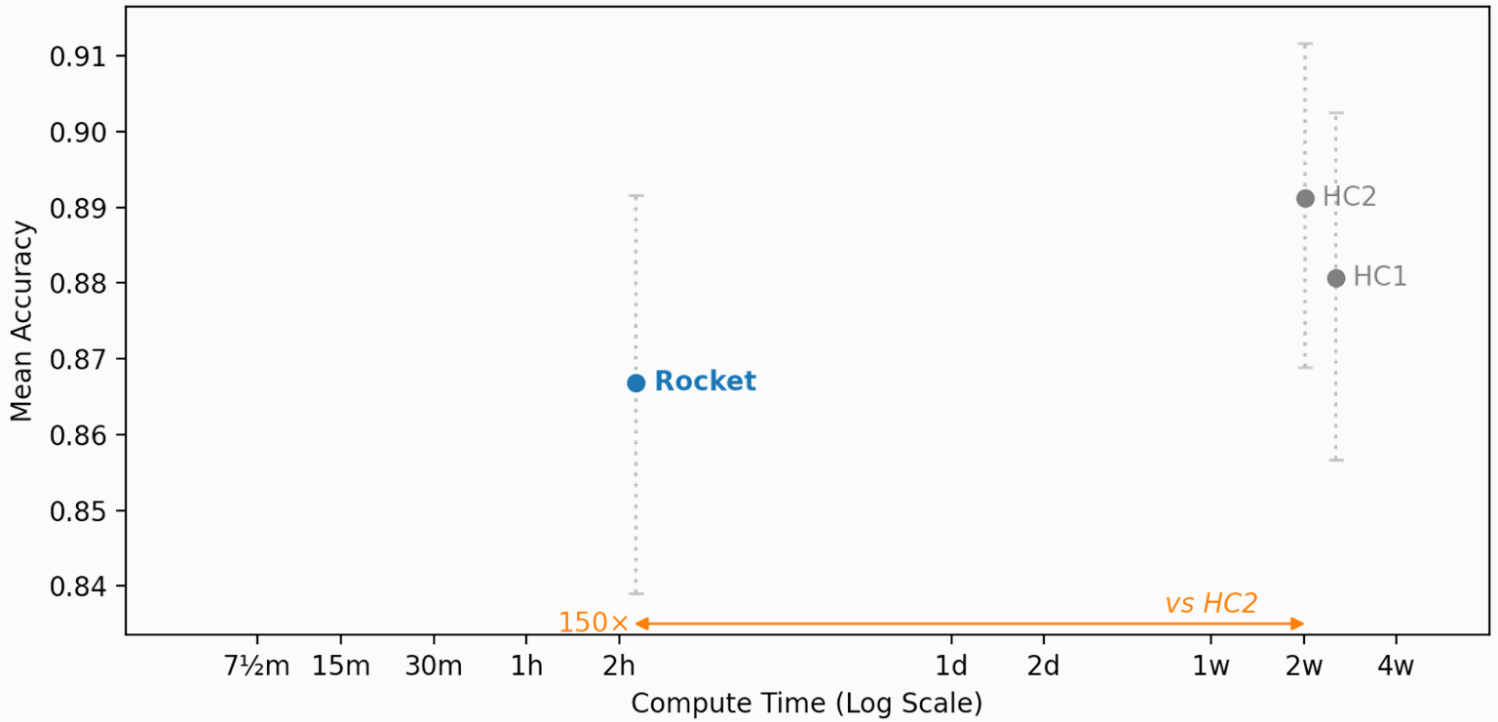
ROCKET: Random choices per kernel

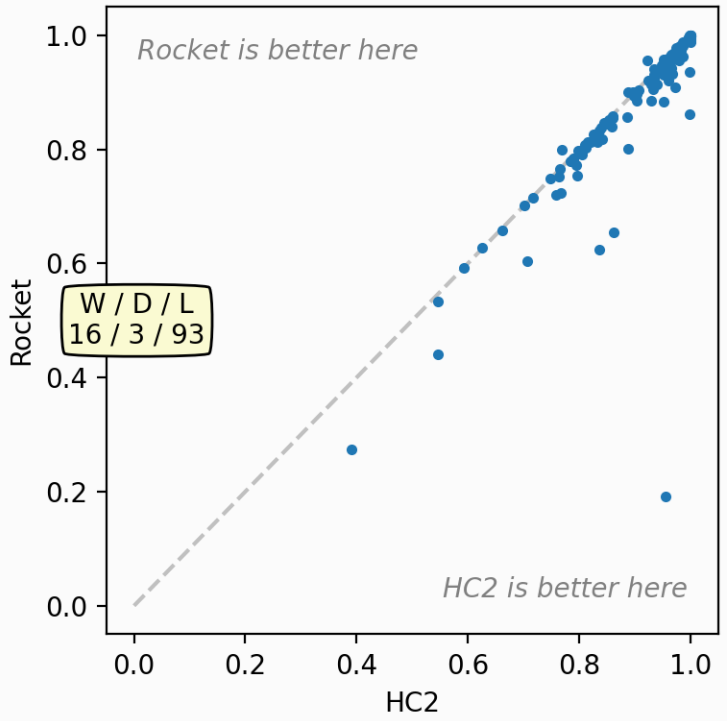
- Length: {7, 9, 11}
- Weights: $N(0, 1)$, then normalized to sum to zero
- Bias: $U(-1, 1)$
- Dilation: $2^{U(0, A)}$, where A limits kernel to series length
- Padding: {T, F}
- Pooling operators: PPV + Max

ROCKET



Rocket

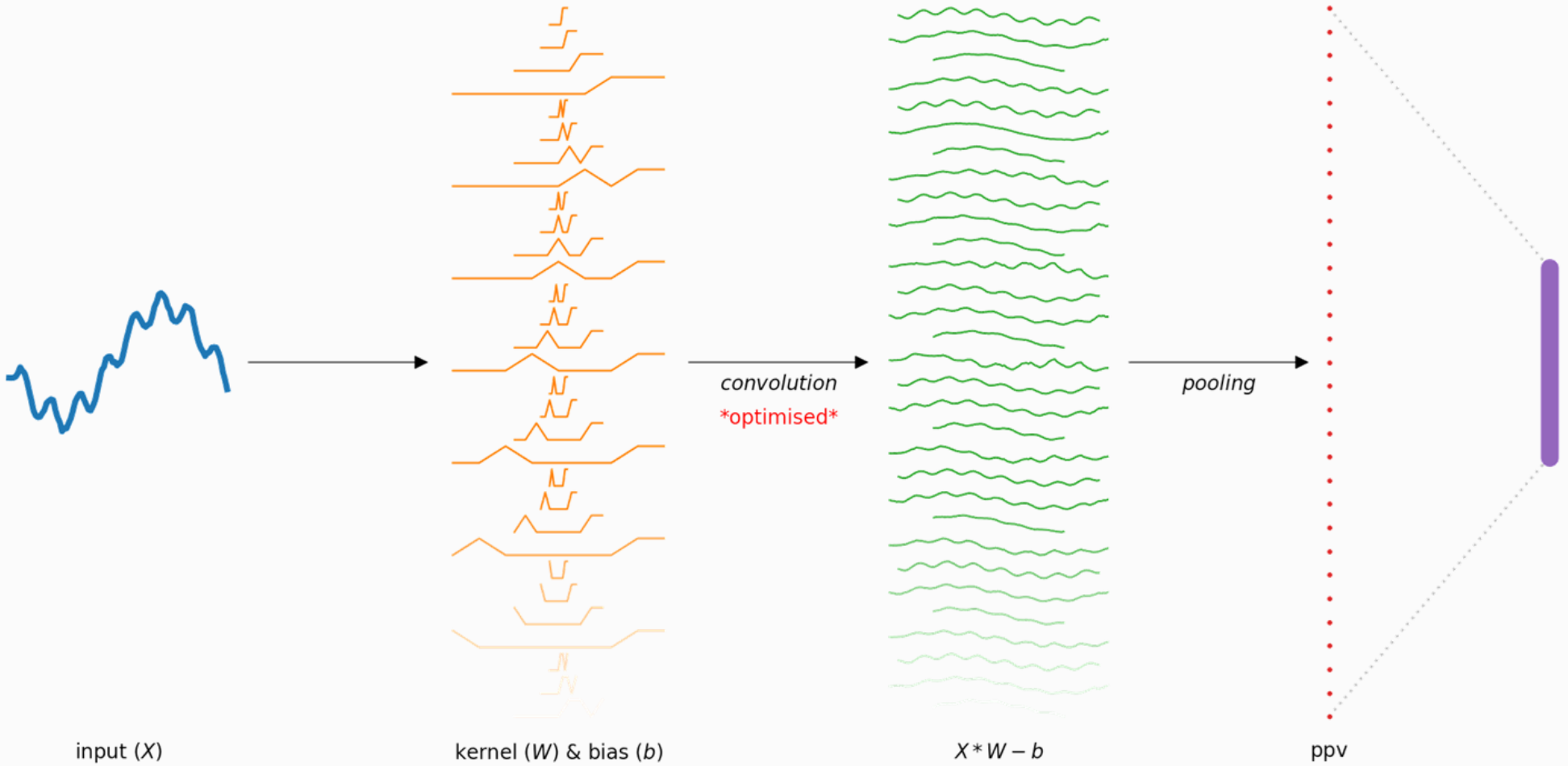

0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
Mean Accuracy
HC2
HC1
Rocket
150×
vs HC2
7½m
15m
30m
1h
2h
1d
2d
1w
2w
4w
Compute Time (Log Scale)


Rocket is better here
W / D / L
16 / 3 / 93
HC2 is better here
1.0
0.8
0.6
0.4
0.2
0.0
Rocket
0.0
0.2
0.4
0.6
0.8
1.0
HC2

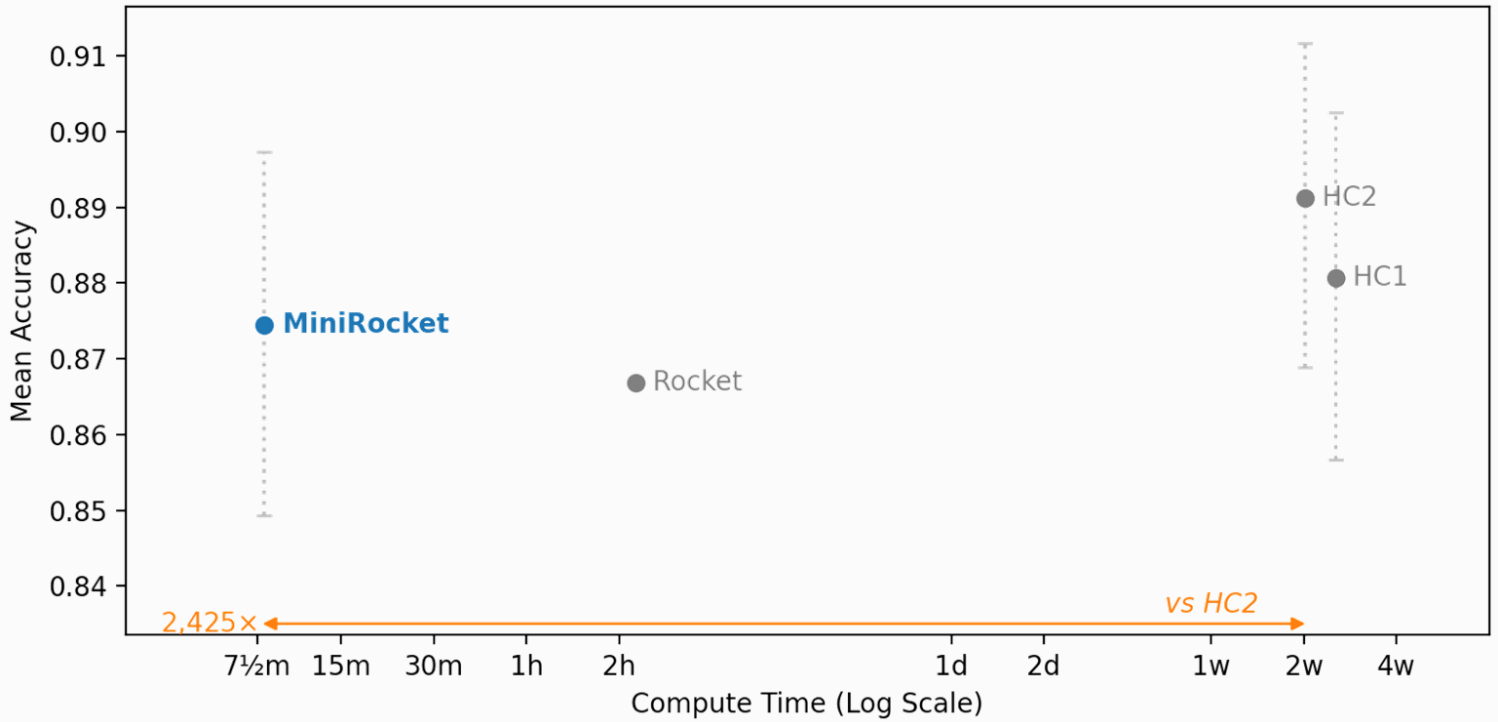
MiniRocket: Fixed choices per kernel

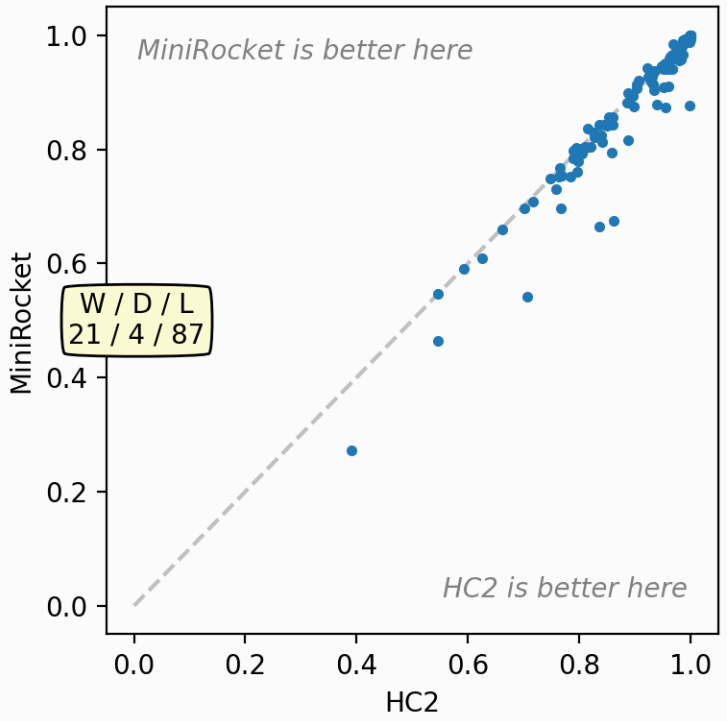
- Length: $\{7, 9, 11\}$
- Weights: $\mathcal{N}(0, 1)$ $\{-1, 2\}$, such that sum to zero
- Bias: $U(-1, 1)$ from convolution output
- Dilation: $2^{U(0, A)}$ fixed (relative to input length)
- Padding: {T, F}
- Pooling: PPV + Max

MiniRocket

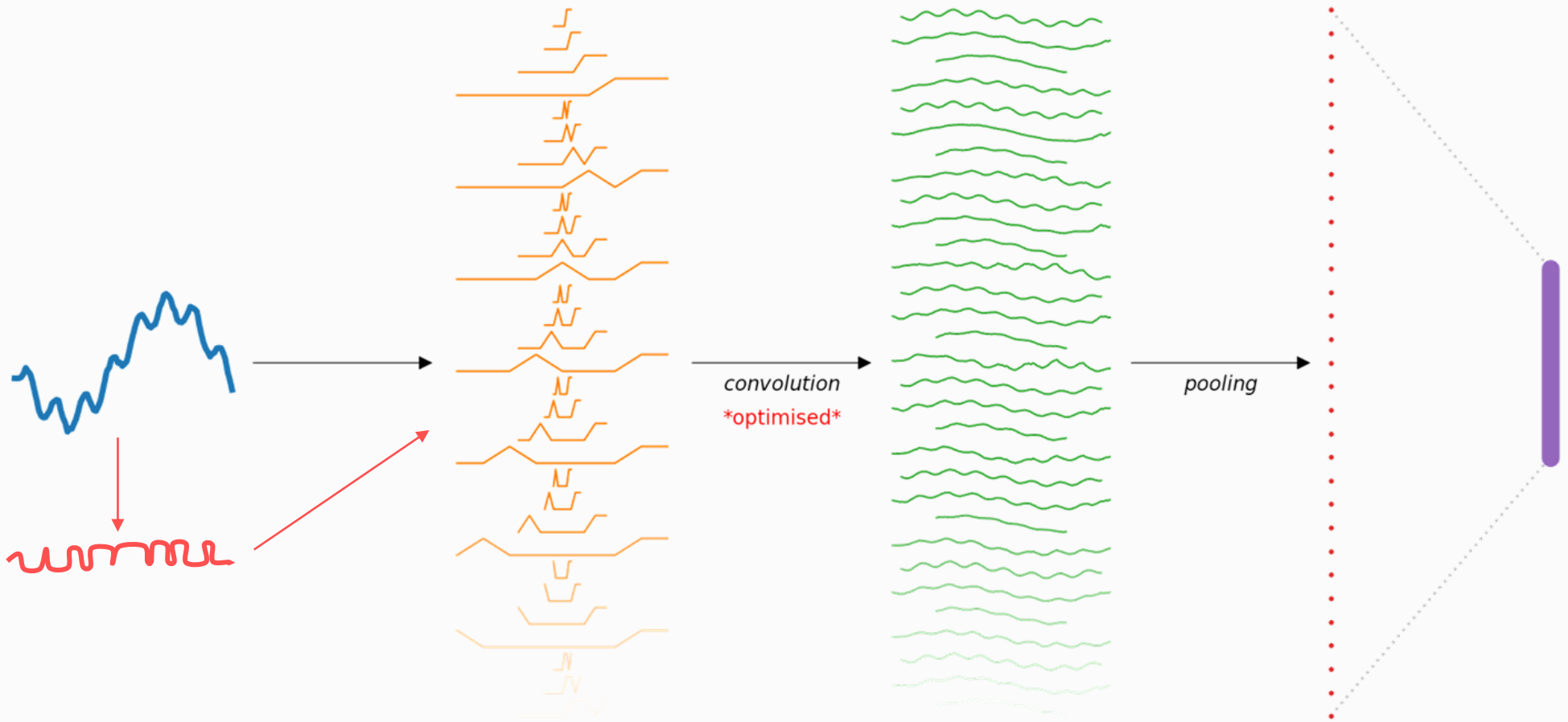

convolution
optimised
pooling
input (X)
kernel (W) & bias (b)
X * W − b
ppv

MiniRocket


0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
Mean Accuracy
MiniRocket
Rocket
HC2
HC1
2,425×
vs HC2
7½m
15m
30m
1h
2h
1d
2d
1w
2w
4w
Compute Time (Log Scale)

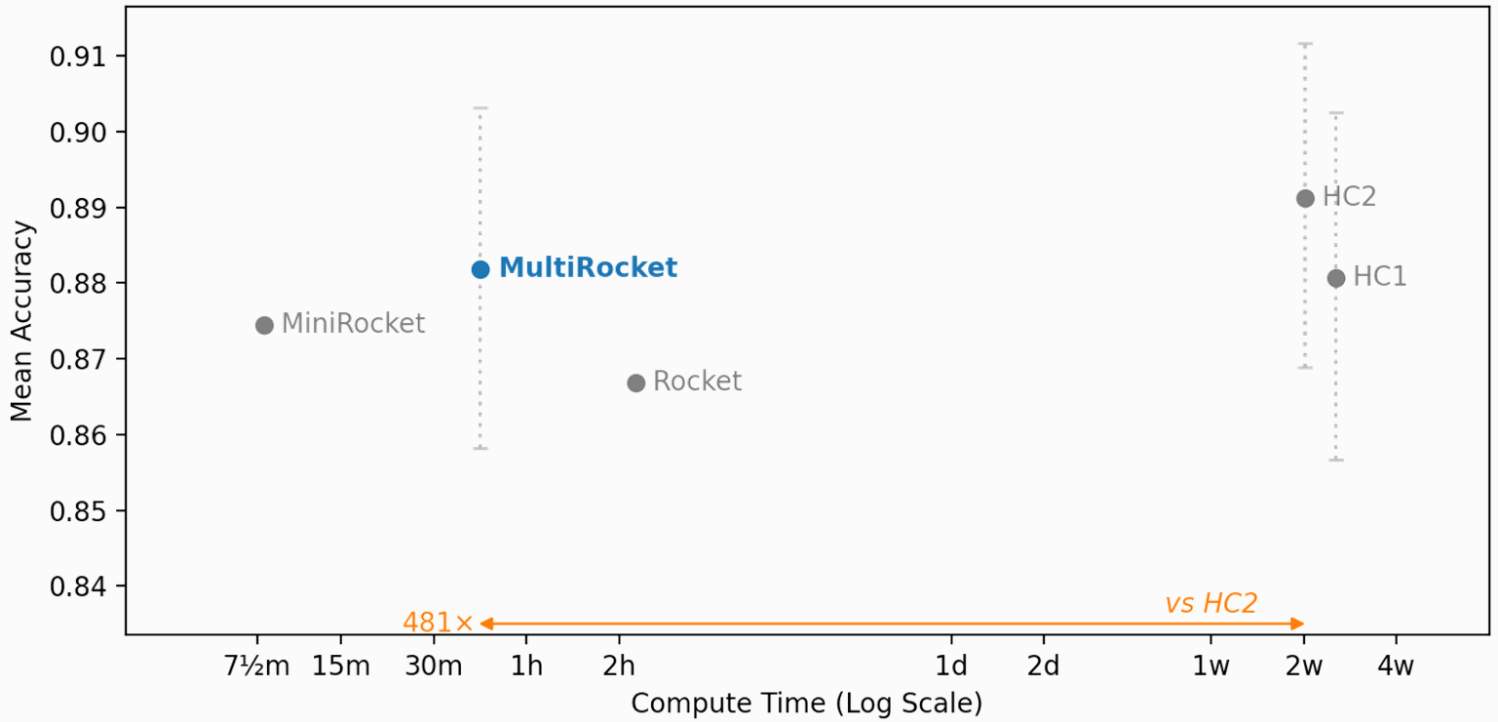

1.0
0.8
0.6
0.4
0.2
0.0
MiniRocket
MiniRocket is better here
W / D / L
21 / 4 / 87
HC2 is better here
0.0
0.2
0.4
0.6
0.8
1.0
HC2

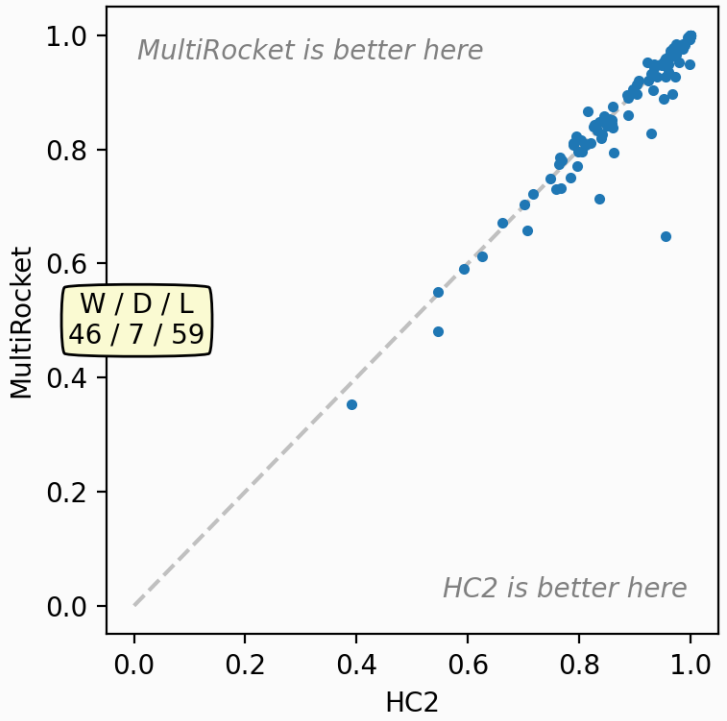
MultiRocket



ppv (how many)
Mean Positive Value (how strong)
Mean Positive Value Index (where)
Longest Stretch of Positive Values (how dispersed)

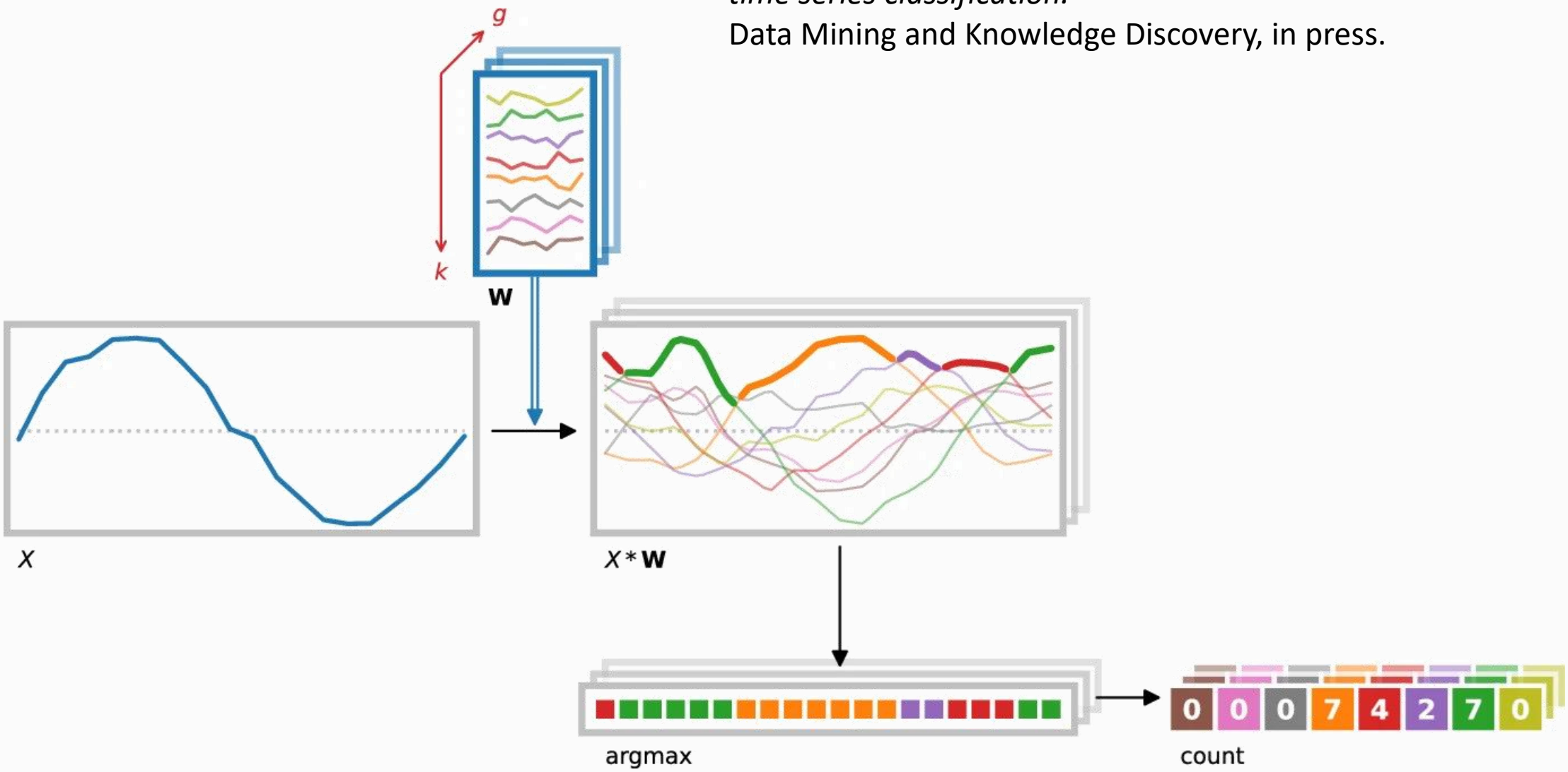
MultiRocket



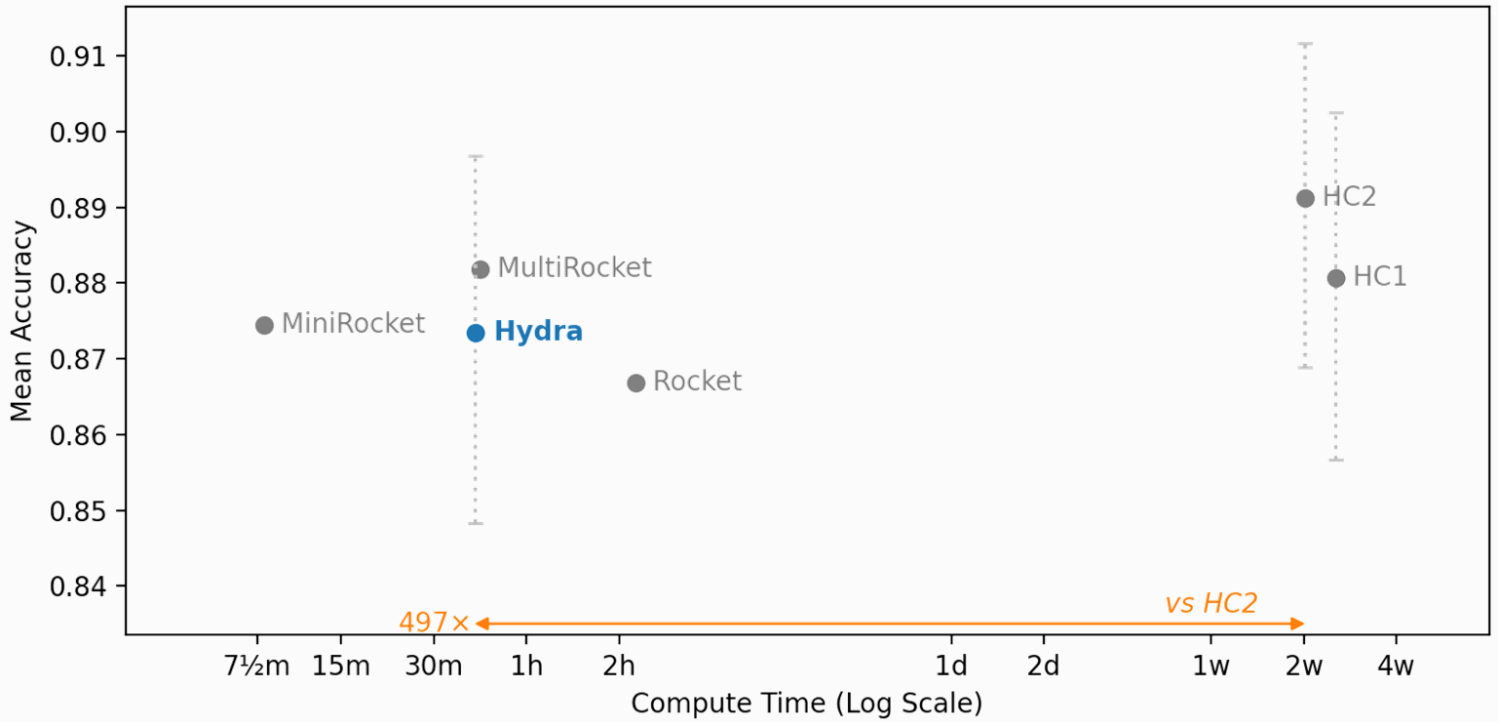


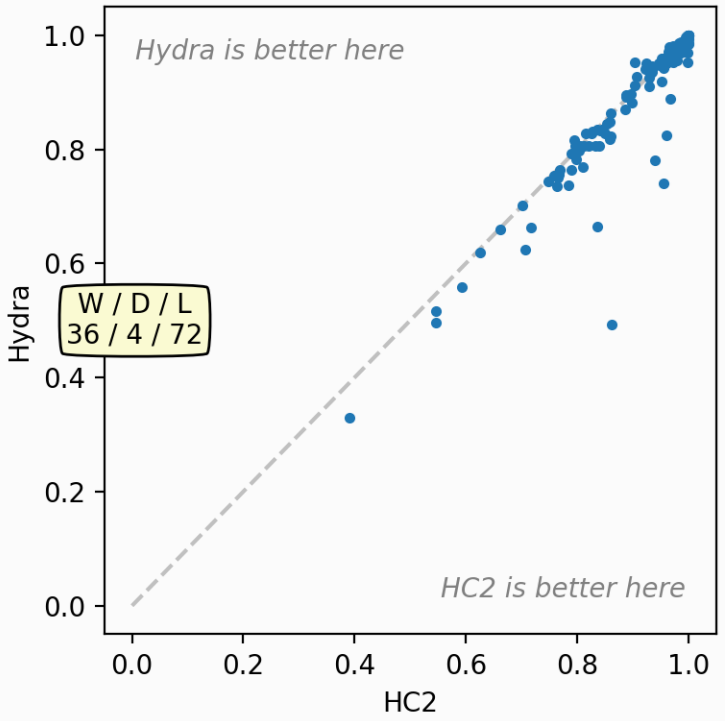
Hydra

Dempster, A., Schmidt, D. F., & Webb, G. I.
HYDRA: competing convolutional kernels for fast and accurate time series classification.
Data Mining and Knowledge Discovery, in press.

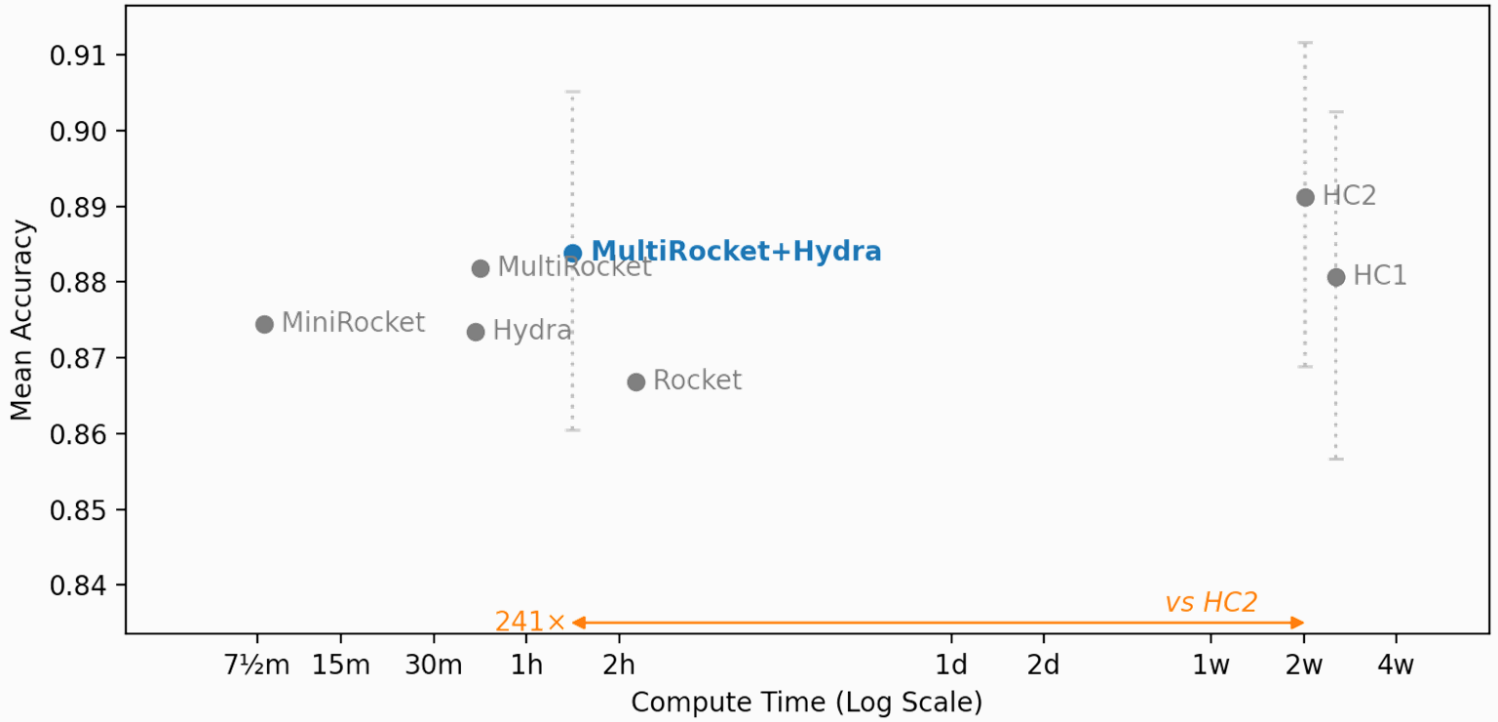


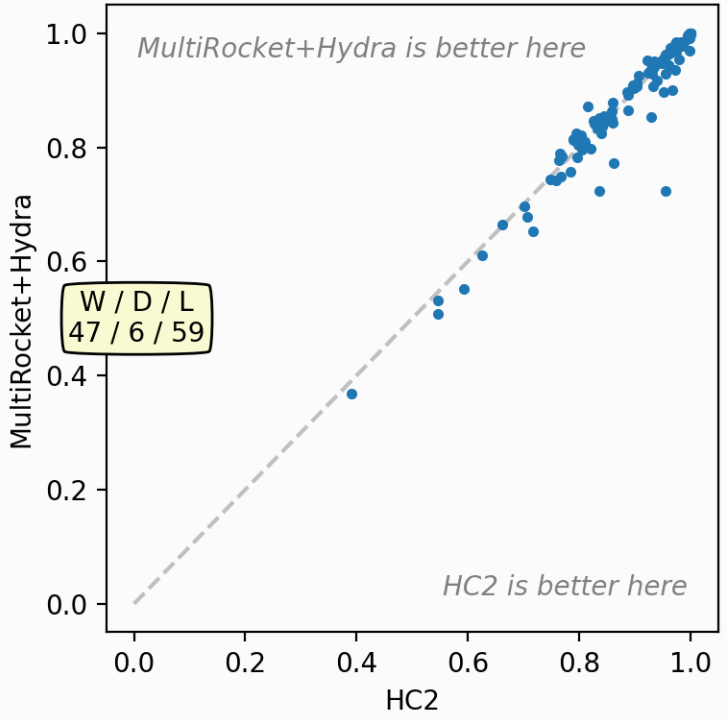
Hydra


0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
Mean Accuracy
MiniRocket
MultiRocket
Hydra
Rocket
HC2
HC1
497×
vs HC2
7½m
15m
30m
1h
2h
1d
2d
1w
2w
4w
Compute Time (Log Scale)

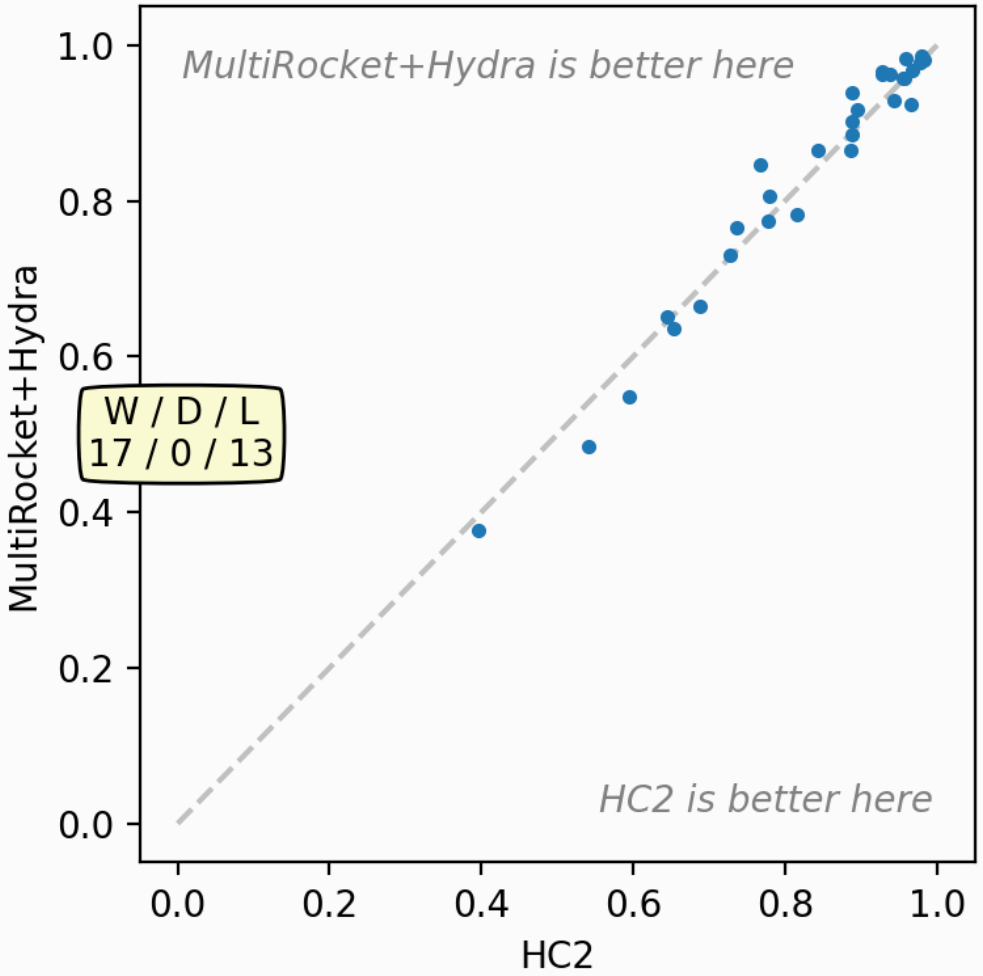

Hydra is better here
HC2 is better here
W / D / L
36 / 4 / 72
1.0
0.8
0.6
0.4
0.2
0.0
Hydra
0.0
0.2
0.4
0.6
0.8
1.0
HC2

MultiRocket+Hydra



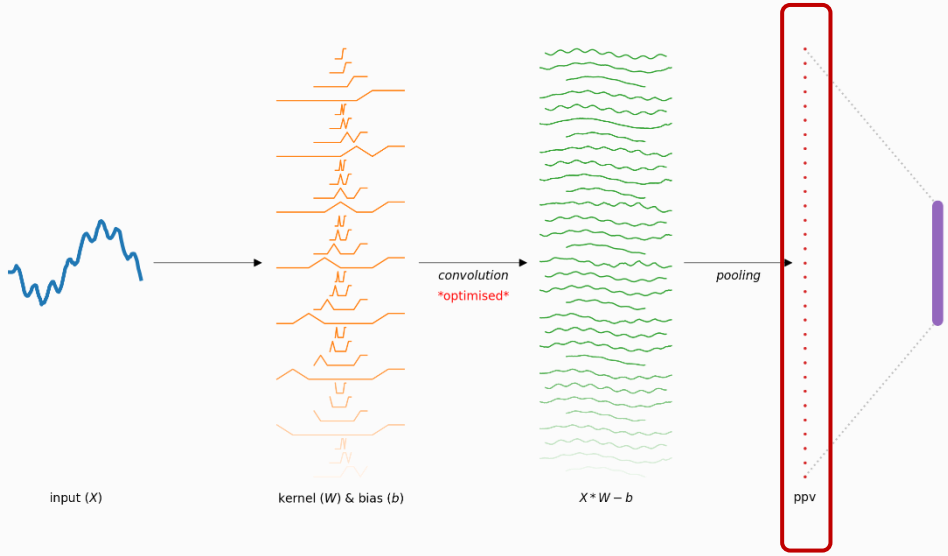


Performance on the 30 new UCR datasets



Research opportunities

- Additional pooling operators



- Employ other forms of kernel
- Use ROCKET features in other time series analytics
  - Forecasting, regression, clustering, anomaly detection, …
- Use non-linear classifiers
- Apply to other data types

Conclusions

- **Rocket** revolutionized time series classification
  - state of the art accuracy
  - many orders of magnitude less computation
- **MiniRocket** achieves another order of magnitude speed up
  - shows that stochasticity does not directly contribute to accuracy
- **MultiRocket** provides substantial gain in accuracy for modest computation
- **Hydra** provides a further consistent gain in accuracy for modest computation
- We believe in reproducible research:
  - **Rocket** → https://github.com/angus924/rocket
  - **MiniRocket** → https://github.com/angus924/minirocket
  - **MultiRocket** → https://github.com/ChangWeiTan/MultiRocket
  - **Hydra** → https://github.com/angus924/hydra

While I've got you!

- **TempCNN** is a leading earth observation analysis method
- **SETAR-Tree** is a powerful global time series forecaster (A7i, 16:30, Tuesday)
- **LB_Enhanced** dominates LB_Keogh on speed and tightness
- **UltraFastMPSearch** gives very fast distance measure meta-parameter tuning
- **ADTW** improves on windowing for DTW
- **Cost function tuning** is a powerful tool for DTW (A11i, 14:20 Wednesday)
- **EAP** greatly speeds up DTW and variants
- **InceptionTime** is the most accurate Deep Learning method on UCR repository
- **ConvTran** is significantly more accurate than InceptionTime on UEA multivariate repository (best journal track papers, 10:30 Wednesday)
- **Proximity Forest 2** is the leading similarity classifier on UCR repository
- **QUANT** shows that all you need is quantiles

Thank you!

Angus Dempster

Geoff Webb

Francois Petitjean

Daniel Schmidt

Chang Wei Tan

http://i.giwebb.com/

Images and video licensed from Shutterstock and Pexel