Topological Simplification of Signals for Inference and Approximate Reconstruction

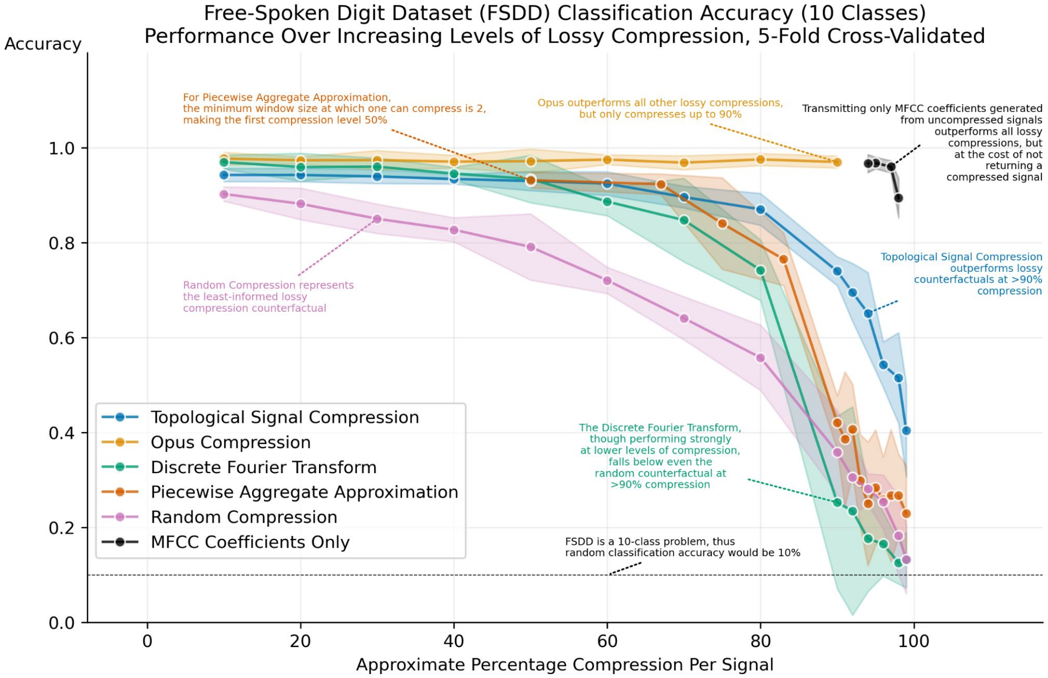
Gary Koplik, Nathan Borggren, Sam Voisin, Gabrielle Angeloro, Jay Hineman, Tessa Johnson, Paul Bendich

March 9th, 2023

GEOMETRIC DATA ANALYTICS, INC.

Takeaways

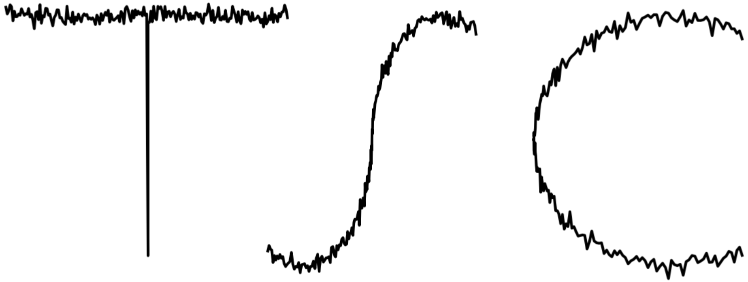
- Topology-inspired lossy compression algorithm: Topological Signal Compression (TSC).
- Generalizable to any 1d signals / sequences of floating point values.
- Interpretable.
- Robust to variable bandwidth constraints.



Code

- `topological-signal-compression` Python package

- Documentation
    - FSDD Example
    - Music (Chopin) Example



Topological Signal Compression

Most Promising TSC Use Case

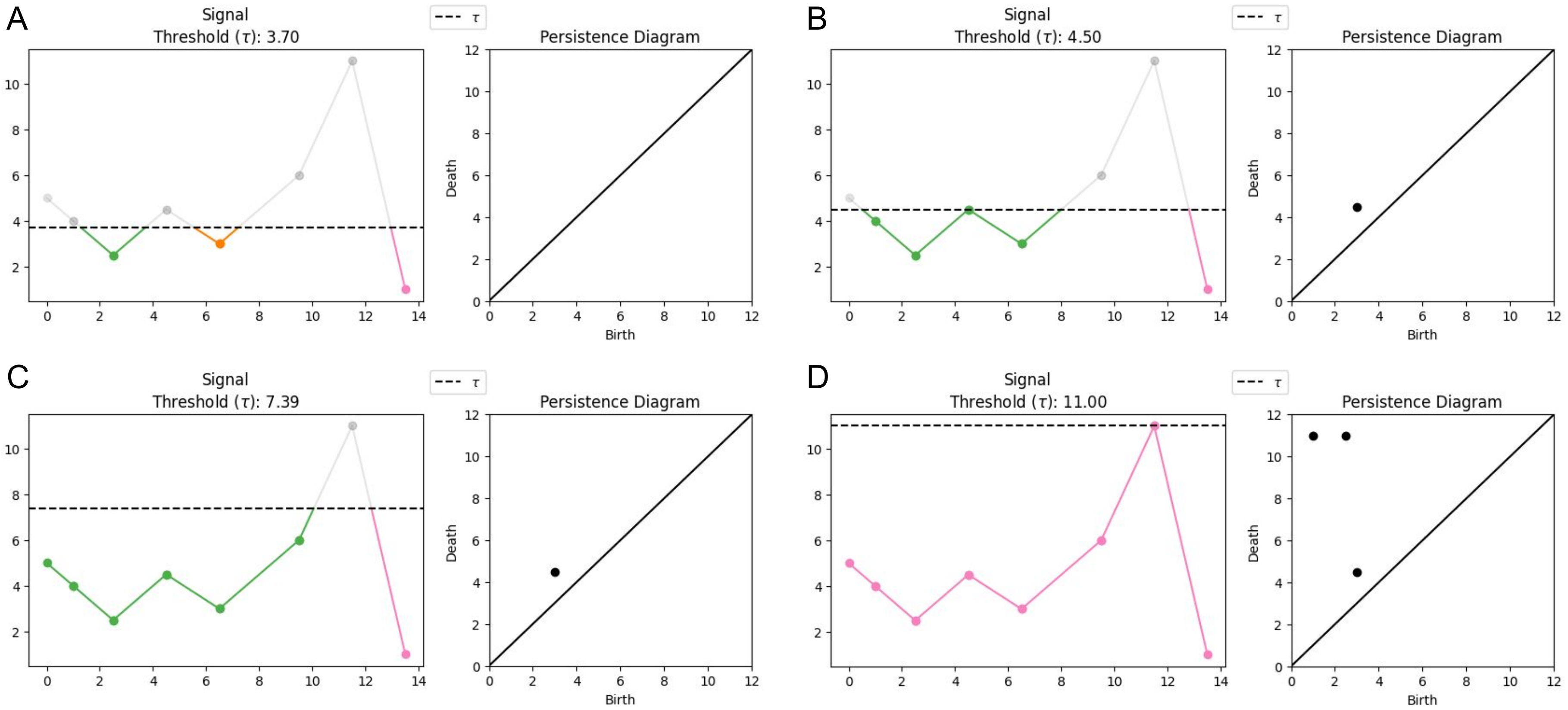
- Internet of Things (IoT) with multimodal sensing capabilities.
    - Generalizability helpful.
- Avoiding black box concerns with highly-interpretable compression.
- Small bandwidth constraint requiring high levels of compression.
    - GDA’s use case: 300 byte message limit with Iridium.
- Variable bandwidth constraint.
    - Relative prioritization of different modalities over time, either via automation or user requests.
    - Needs to be stable to downstream ML.
- Preserving exact values of critical points.
    - Desire to send data points rather than summary statistics or featurizations.

Overview

- Topological Data Analysis (TDA) Background.
  - Persistent homology and its application to signals.
- Topological Signal Compression (TSC).
  - Brief mathematical justification and a quick example.
- Free-Spoken Digit Dataset (FSDD).
- Machine Learning.
- Counterfactual Lossy Compression Algorithms.
- Results.
- Theoretical Benefits of TSC.
  - Contrast TSC with counterfactual compression algorithms.
- Future Work.

TDA Background

Persistent Homology and Persistence Diagrams

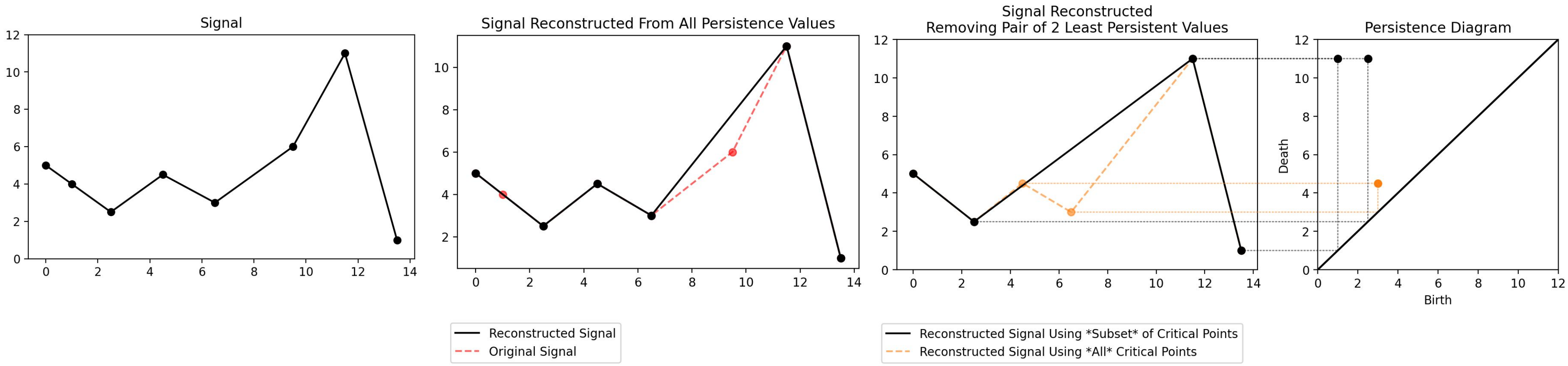


Topological Signal Compression (TSC)

Topological Signal Compression (TSC)

- Persistent homology thus gives us an *ordering* on our connected components. Dots on the persistence diagram that are closer to the diagonal die sooner after being born.

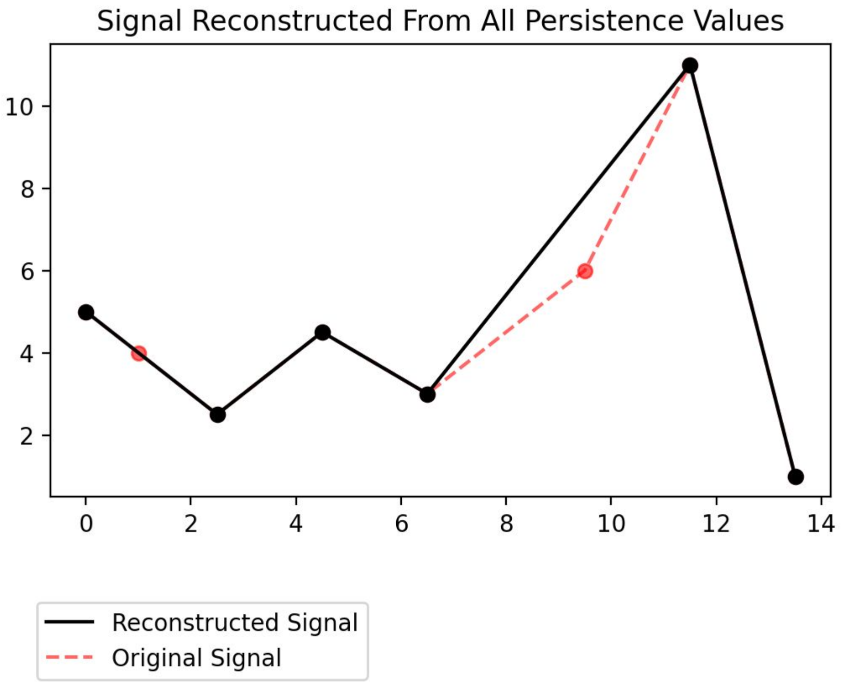
- TSC enables the reconstruction of the signal by *removing* the critical points (local mins and maxes) of least persistence, keeping the more prominent features of the signal.



Minimum Compression with TSC

- In tracking the births and deaths of components, we only track the *critical points* in a signal.

- The minimum compression performed by TSC is to drop all non-critical points.

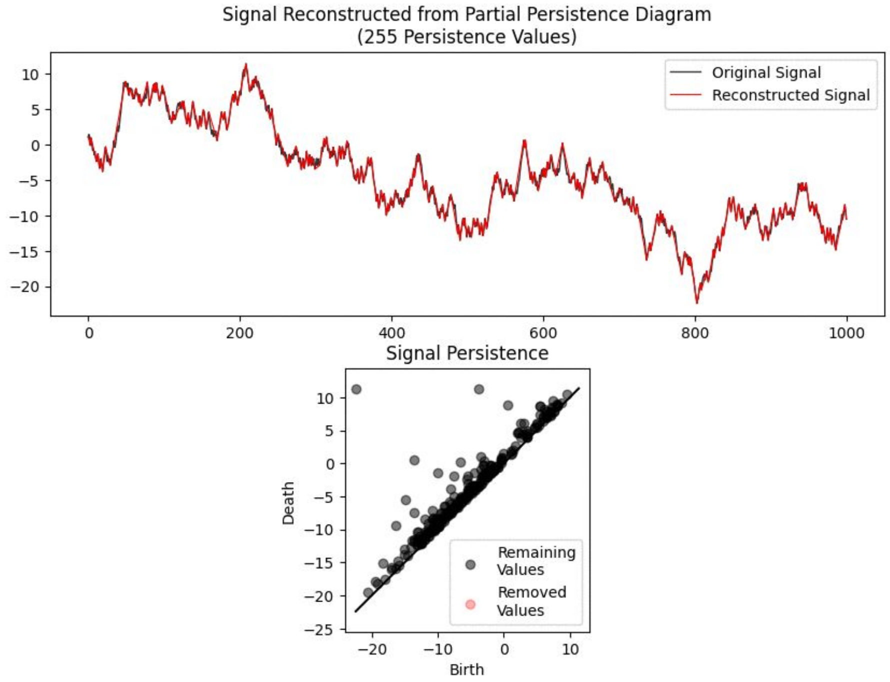


Mathematical Basis of TSC

- The *Morse Cancellation Lemma (MCL)* guarantees that the two values from the signal corresponding to the dot of lowest persistence are *horizontally adjacent* critical points in the time domain.

- This guarantees that *removing* this pair of points will not disrupt any other pair, thus preserving the critical point information over the rest of the signal.

- MCL can be applied iteratively to achieve arbitrary levels of compression, as demonstrated in the figure to the right.



Free-Spoken Digit Dataset (FSDD)

FSDD Summary

- Audio dataset produced by six male speakers consisting of recordings of spoken digits 0 through 9.

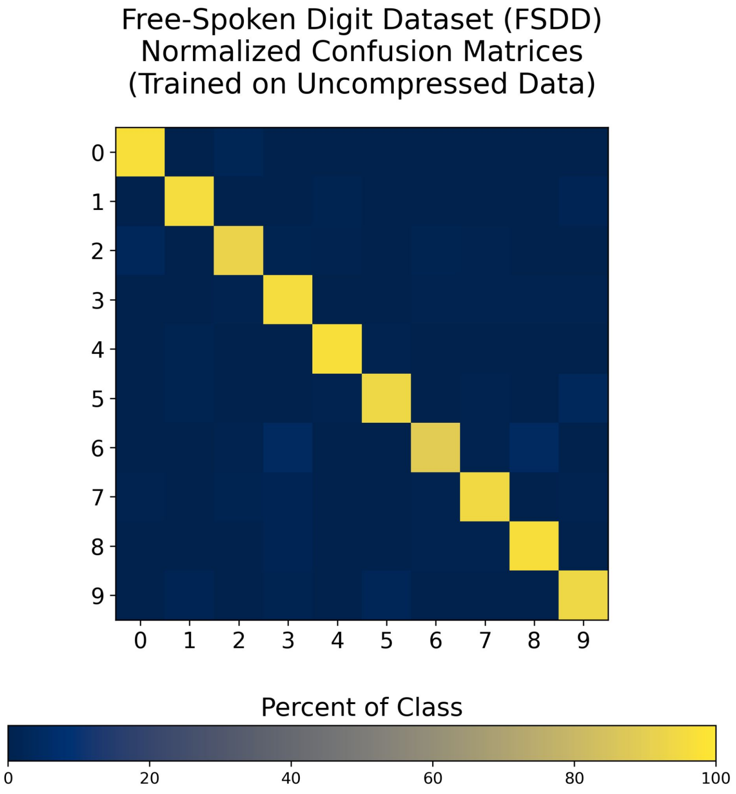
- 3000 examples recorded at 8 kHz, with each person recording 50 samples of each digit.

Machine Learning

Baseline Machine Learning Exercise

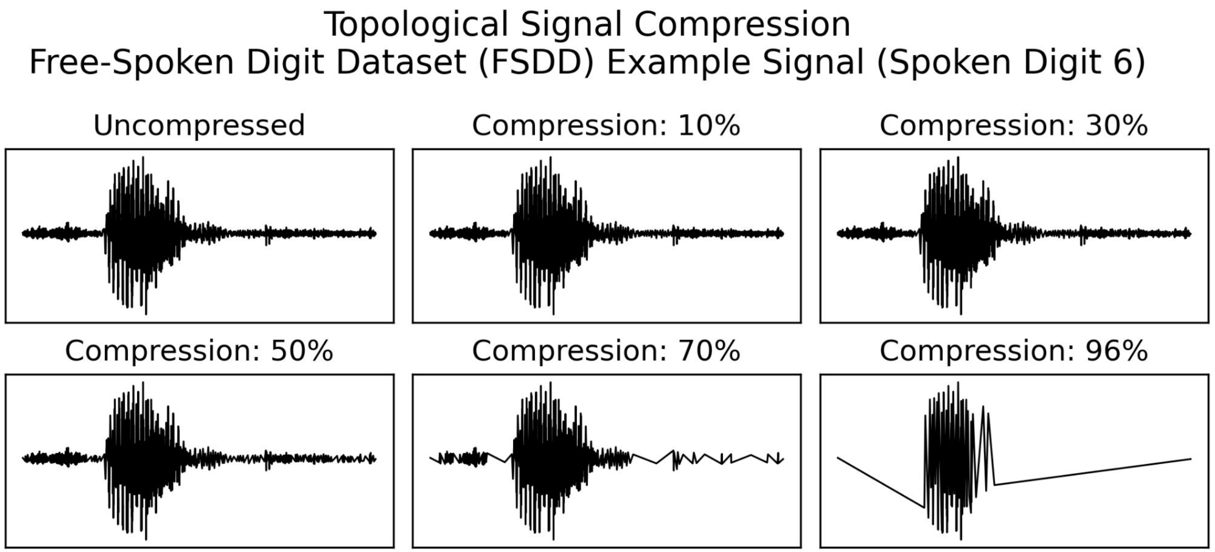
- Classified FSDD spoken digits using:
  - Mel-Frequency Cepstrum Coefficient (MFCC) features.
  - Run through a Convolutional Neural Network (CNN).
  - With 5-fold cross-validation.

- For this initial exercise, mean cross-validated accuracy of 97%.



Machine Learning on Compressed Data

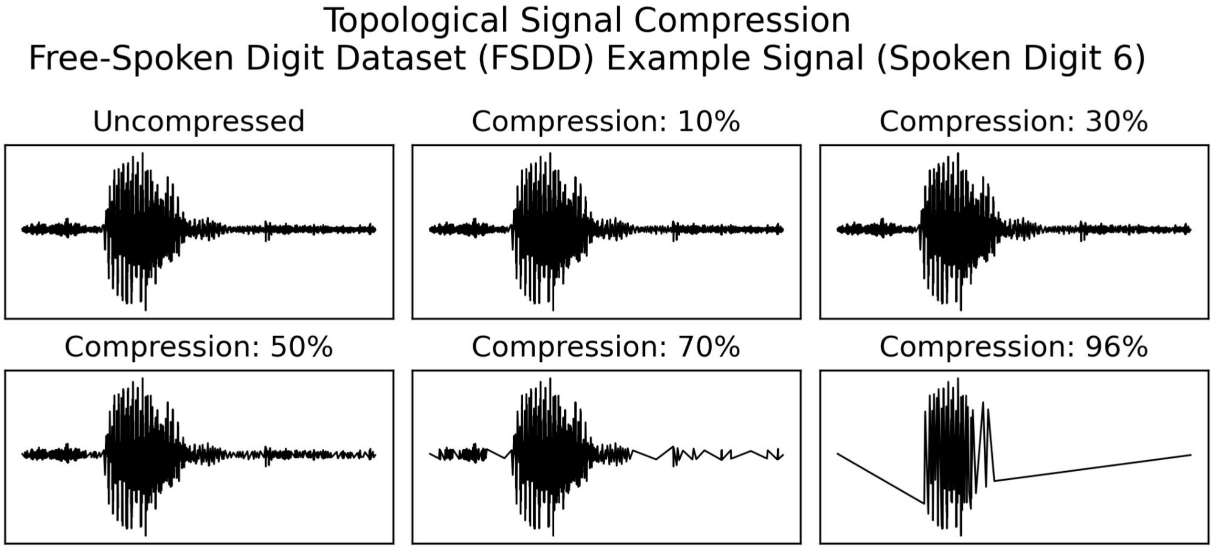
- Repeat this cross-validated classification pipeline, instead building the MFCC features using increasingly compressed signals from FSDD.



Counterfactual Lossy Compression Algorithms

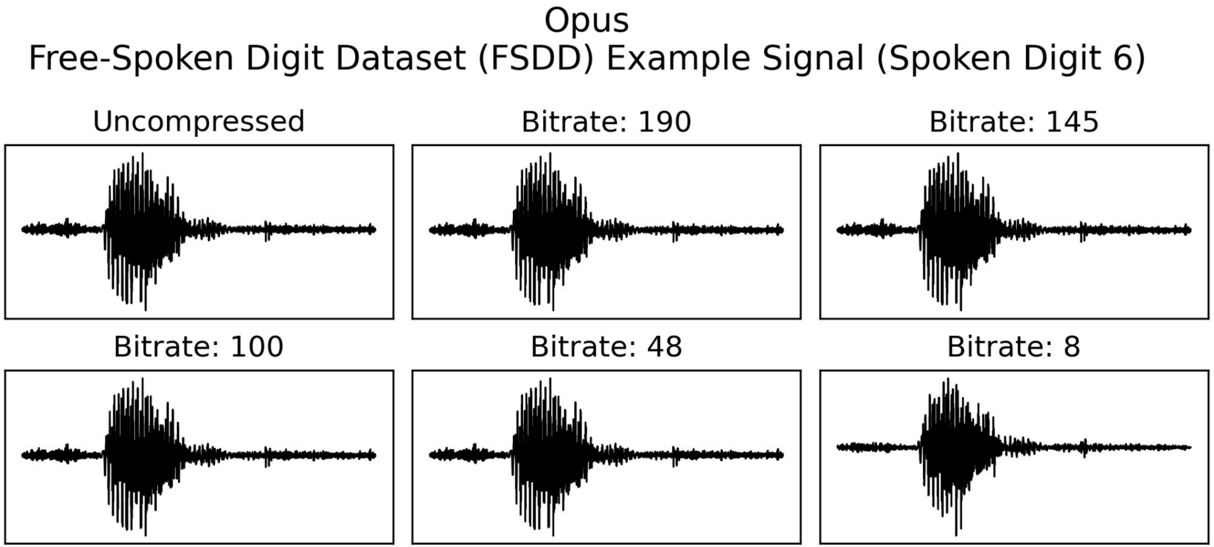
Our Algorithm: TSC

- Dropping data corresponding to data pairs with the lowest persistence values.
- More persistence values dropped ⇒ more compression.



Opus

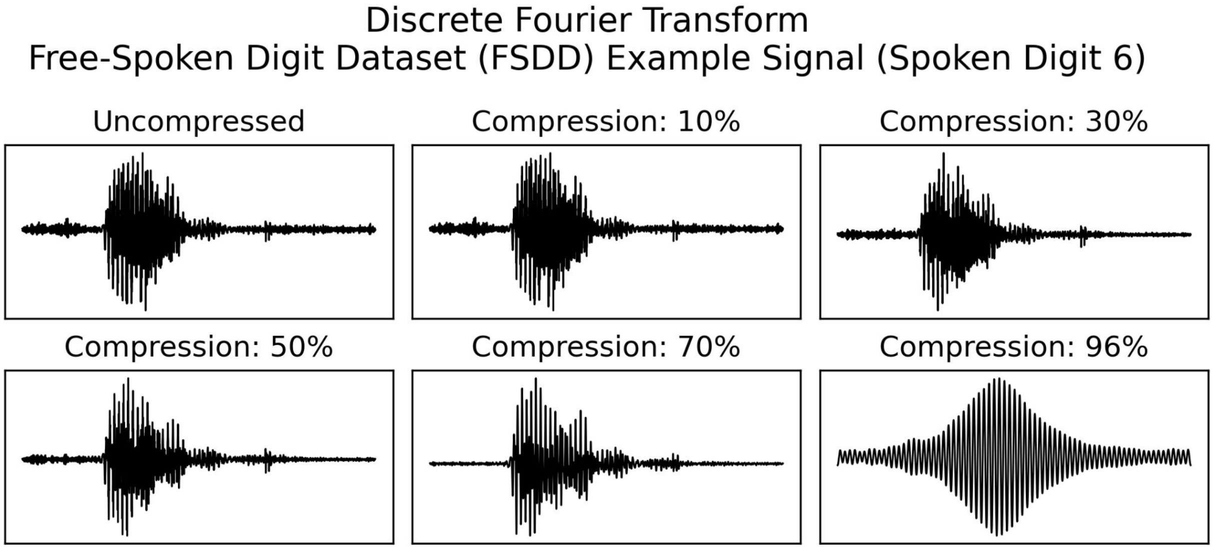
- Codec designed exclusively for lossy audio compression.
- Lower bitrate ⇒ more compression.



Discrete Fourier Transform (DFT)

- Represent a signal with its Fourier coefficients, sending only a subset of those coefficients to reconstruct the signal.

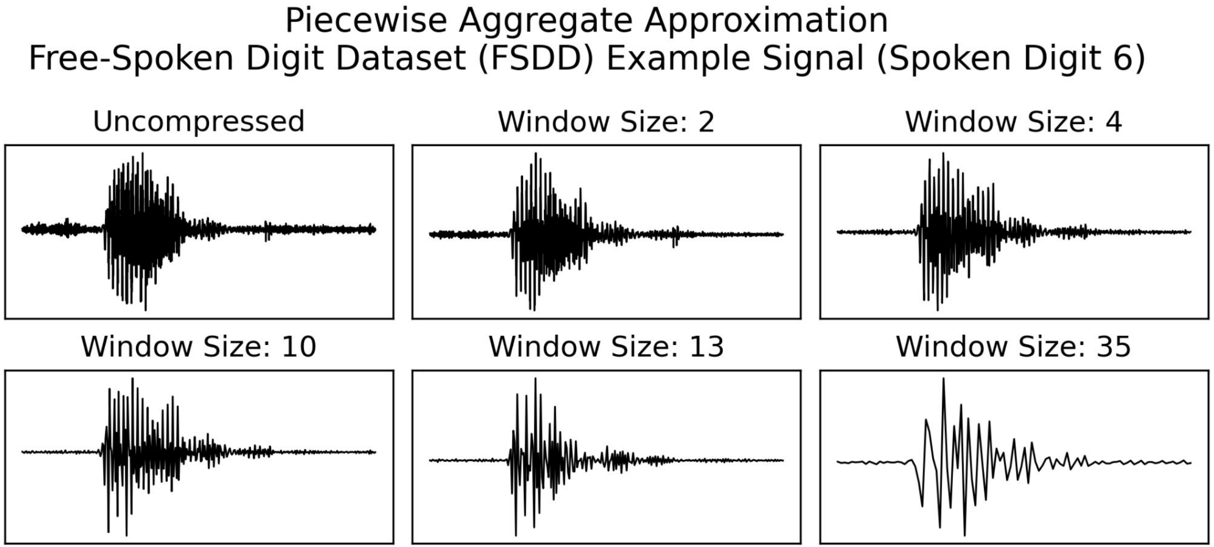
- More Fourier coefficients dropped ⇒ more compression.



Piecewise Aggregate Approximation (PAA)

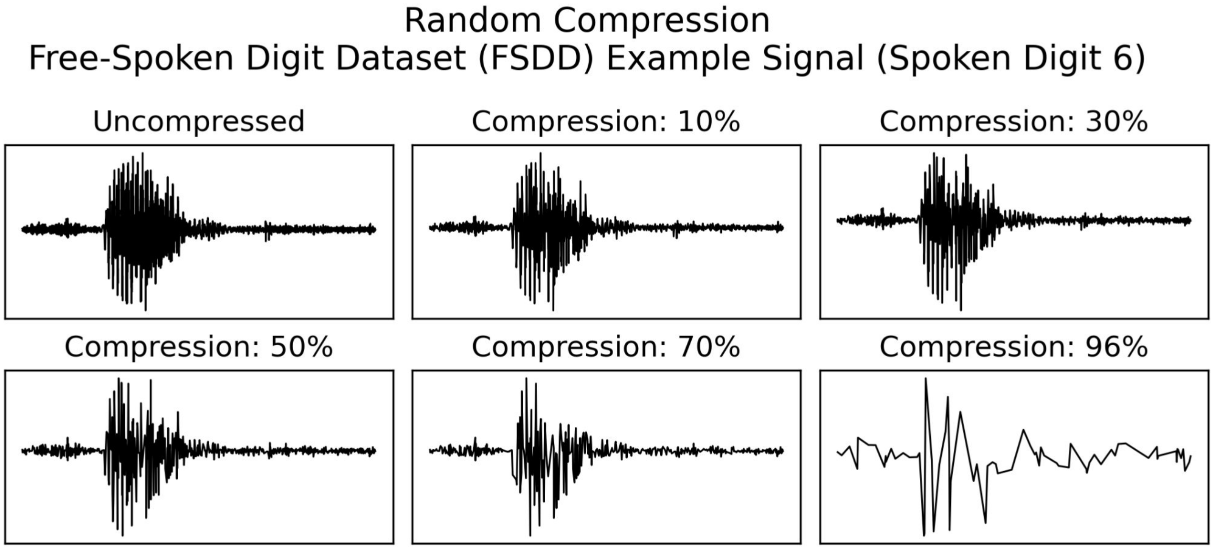
- Uniformly partitions a signal using a fixed window size and returns the mean function value within each window.

- Larger fixed window size ⇒ more compression.



Random Compression

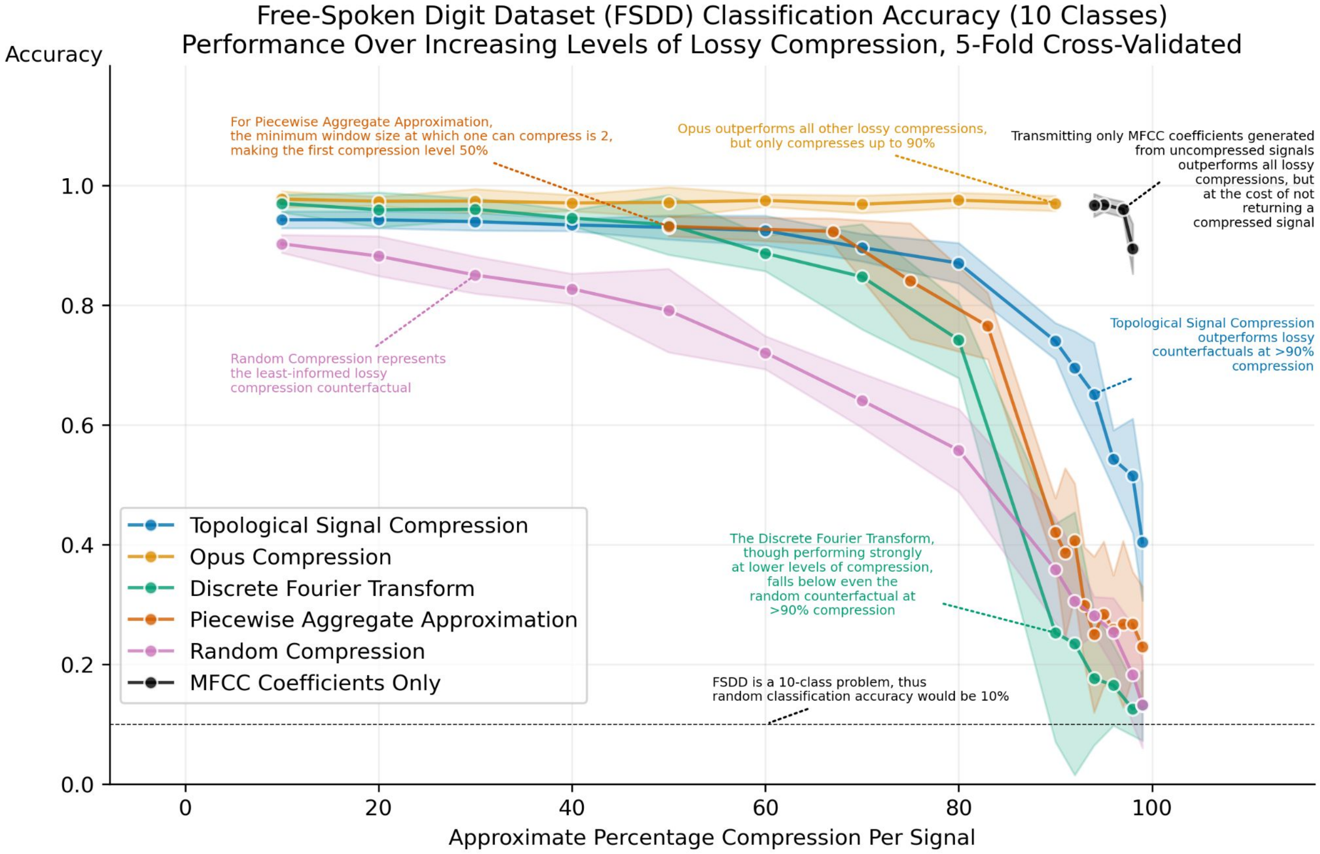
- Random sample (without replacement) of signal values.
- Less values sampled ⇒ more compression.



Results

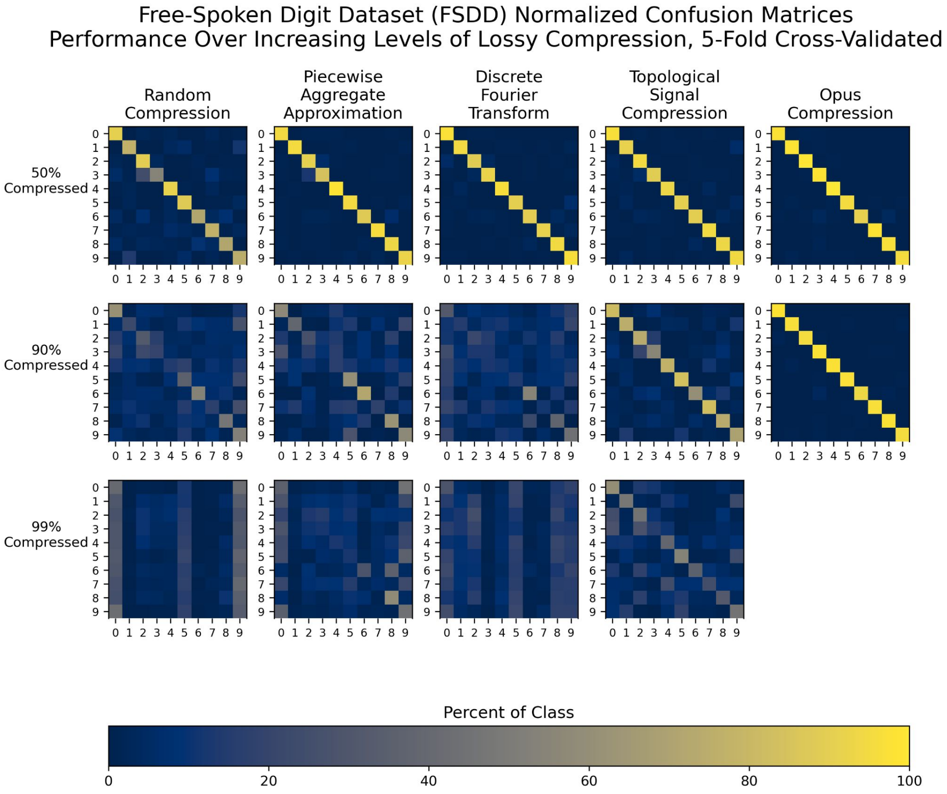
ML Accuracy v. Compression Percentage

- Opus best, but only compresses up to 90% on average on FSDD.
- TSC outperforms other counterfactuals at >90%.



Confusion Matrices

- TSC maintains more consistent classification accuracy over each digit than the non-Opus counterfactual compression algorithms.



Theoretical Benefits of TSC

Generalizability

- Paper explores audio data, but TSC could be used on any other 1d signal.
    - Not the case for Opus.

Interpretability

- A marginal compression with TSC creates a visually-interpretable change in the compressed signal.
  - Not the case for DFT, Opus.

Localized, On-the-Margin Compression

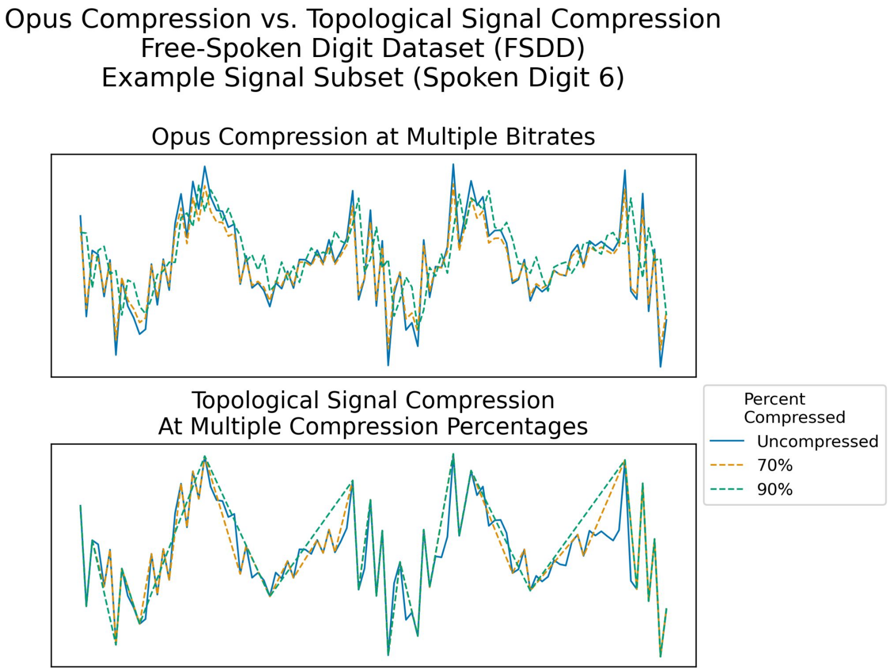
- Only TSC and DFT capable of byte-specific marginal compression.
  - Critical for using full bandwidth constraints.
  - Not the case for Opus and PAA.

- TSC only algorithm with *localized* marginal compression.
  - Not the case for DFT (or Opus, PAA).

- TSC likely more stable for downstream machine learning on variably-compressed signals.

Preserving Exact Values in Signals

- If exact signal values matter, TSC will preserve the exact critical points kept from a signal.
  - Not the case for Opus (see figure on right), DFT, or PAA.



Future Work

Future Work

- Look at more modalities of data.

- More exploration into theoretical robustness of downstream ML to variably-compressed signals.

Thank You

Appendix

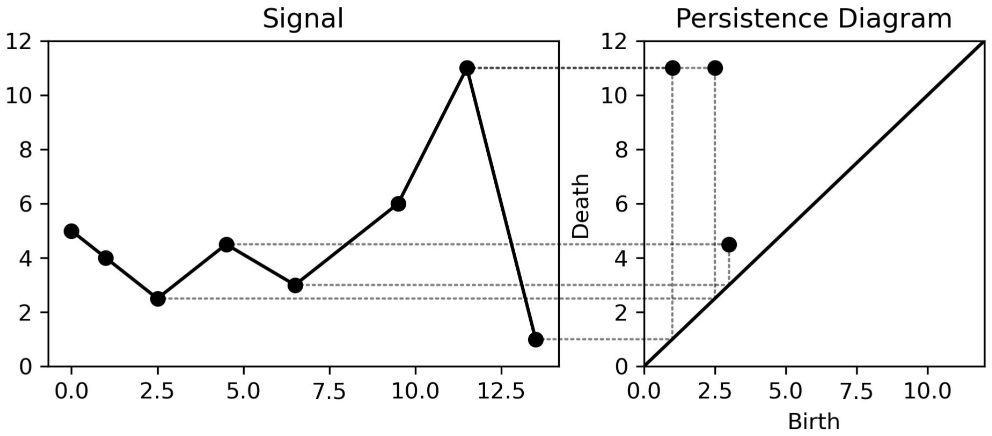
Persistent Homology and Persistence Diagrams

- *Persistent Homology* tracks connected components as we sweep a horizontal line vertically from negative infinity to positive infinity.

- Components are born at local minima, and die at local maxima, destroying the more recently born component of the two components merging.

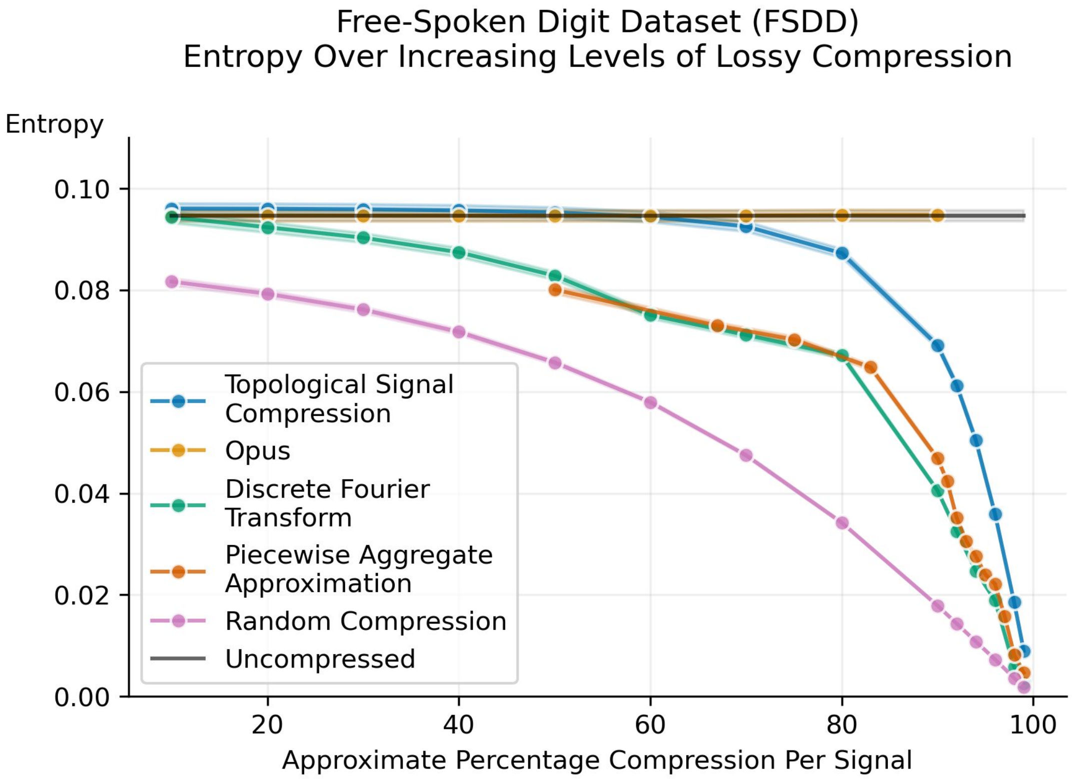
- The *Persistence Diagram* $D_0(f)$ plots the births and deaths of components as dots in the plane.

- The vertical distance of a dot above the 45 degree line *birth = death* represents the *persistence* of a component.



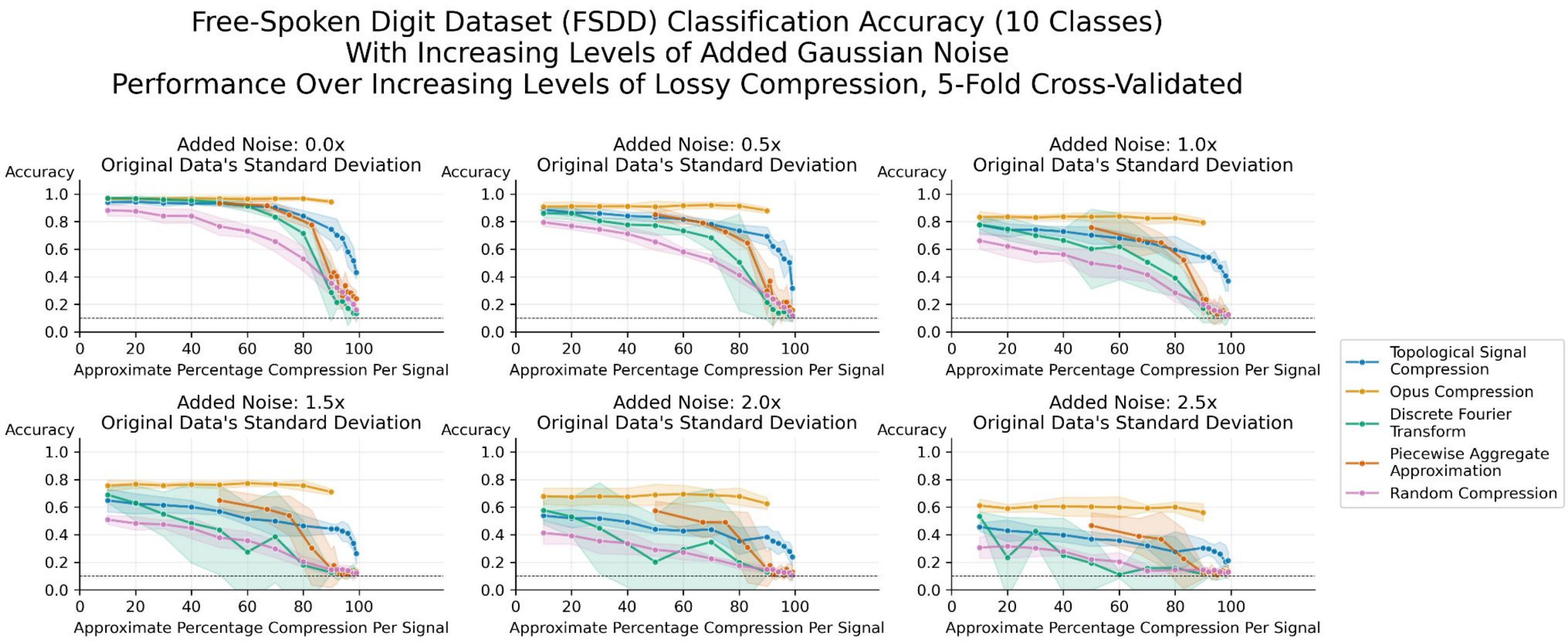
Signal Entropy v. Compression Percentage

- Entropy results comparable to ML results.



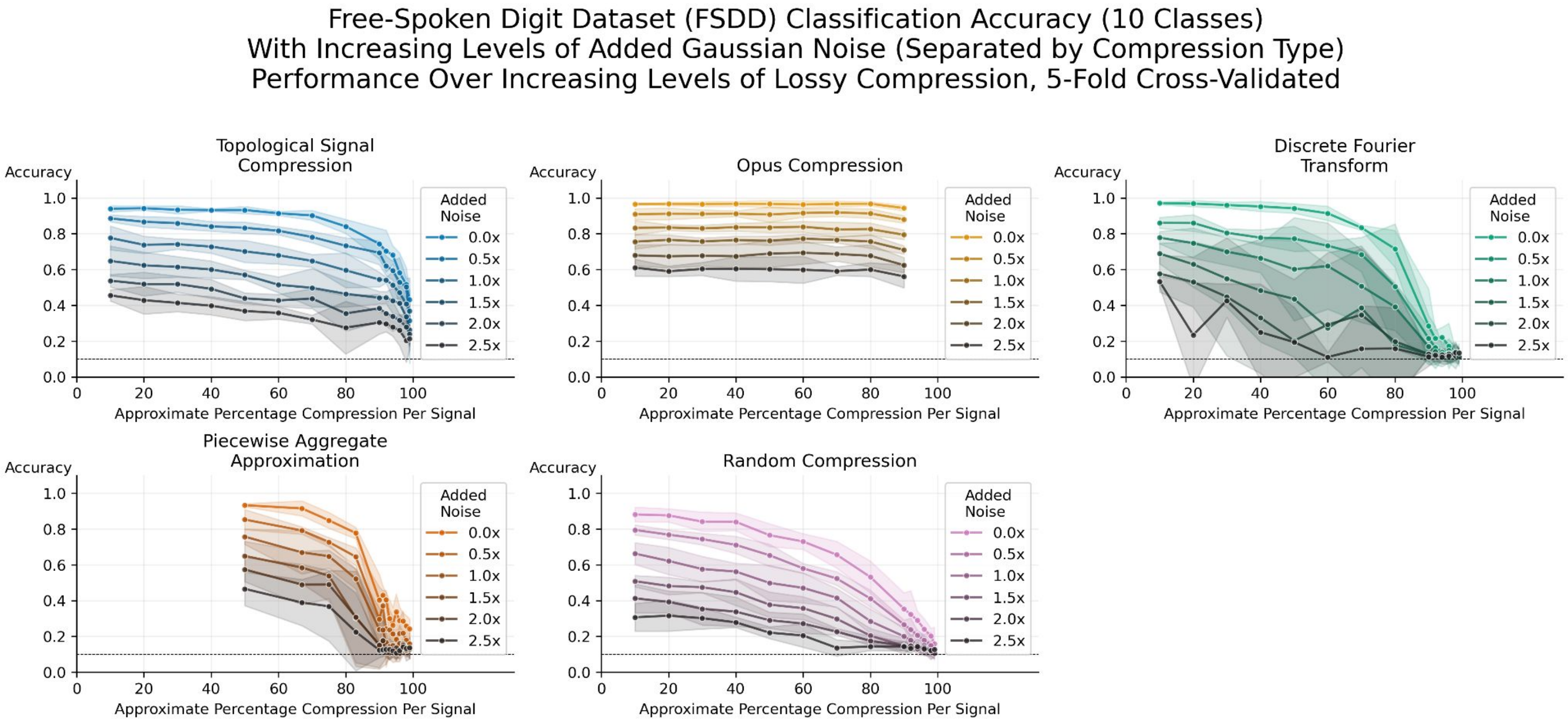
ML Accuracy v. Compression Percentage with Noise

- TSC's relative performance preserved at high compression levels with added noise.



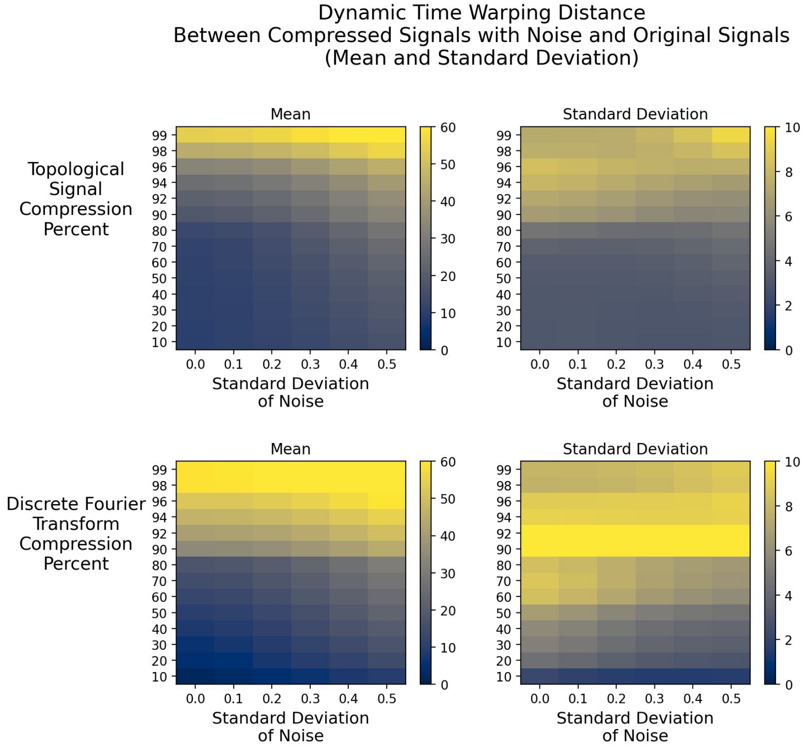
ML Accuracy v. Compression Percentage with Noise

- TSC's declining accuracy smooth over increasing noise.



ML Accuracy v. Compression Percentage with Noise

- TSC's declining accuracy smooth over increasing noise.



Minimum Compression Percentage with TSC

- Values greater than 100% result from needing to store (t, f(t)) values with TSC instead of just f(t) values.