UTILIZING EXPERT FEATURES FOR CONTRASTIVE LEARNING OF TIME-SERIES REPRESENTATIONS

ICML 2022

MANUEL NONNENMACHER, LUKAS OLDENBURG, INGO STEINWART, DAVID REEB

BOSCH CENTER FOR ARTIFICIAL INTELLIGENCE (BCAI), RENNINGEN, GERMANY

UNIVERSITY OF STUTTGART, GERMANY

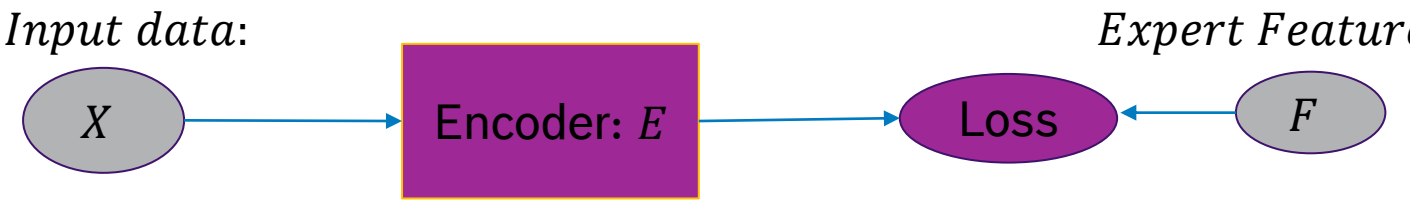
BOSCH

Introduction and Setting

Problem:

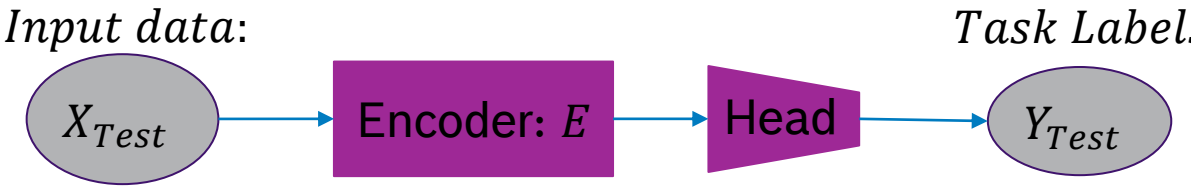
Utilize expert features to learn a representation, which can subsequently be used for a multitude of tasks (e.g., prediction, outlier detection, active learning, transfer learning,…)

Training:



$$\mathrm{X} \in \mathbb{R}^{N \times T \times c}$$
$$\mathrm{F} \in \mathbb{R}^{N \times f}$$
$$\mathrm{E(X)} \in \mathbb{R}^{N \times e}$$

Testing:



$$\mathrm{X_{Test}} \in \mathbb{R}^{N_t \times T \times c}$$
$$\mathrm{Y_{Test}} \in \mathbb{R}^{N_t \times k}$$

Examples:

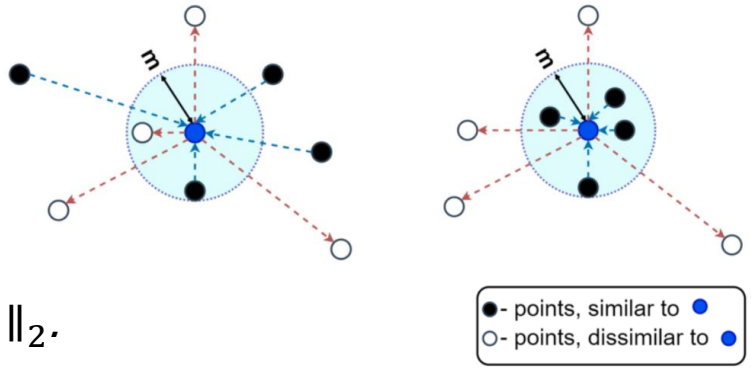
- Input Time-Series: Snippets with GPS location, Speed, Engine Temperature, ….
- Expert Features: Average Speed, Cumulative Elevation Gain, Simulated Data, Number of Peaks, Expert Labels, ….
- Labels: Emissions, Fuel Consumption, Quality Score, Outlier labels, …

Introduction to Contrastive Learning of Representations
The Pair-Loss Function

- **Goal:** Contrastive learning tries to find a representation such that “similar” (same class) data points are close to each other, and “dissimilar” (different class) data points are well separated.

- **Pair-loss:** For two pairs (x_i, y_i) and (x_j, y_j) one can define the pair loss:

$$L_{pair}(x_i, x_j, s_{ij}) = s_{ij} D_{ij}^2 + (1 - s_{ij}) \max(0, m^2 - D_{ij}^2),$$



where $s_{ij} = \delta_{y_i, y_j}$ *(“similarity”) and* $D_{ij} = \| E(x_i) - E(x_j) \|_2$.

Contrastive Learning with Continuous Features

- **Our Idea:** Transferring the idea of contrastive learning to continuous labels/features leads to two desired properties:
  - (P1) If $\| f_i - f_j \|_2$ is small, then $\| E(x_i) - E(x_j) \|_2$ should also be small.
  - (P2) If $\| f_i - f_j \|_2$ is large, then $\| E(x_i) - E(x_j) \|_2$ should also be large.

- **Ansatz:** Extend the discrete similarity measure in natural way to (continuous) expert features:

$$s_{ij} = \delta_{y_i, y_j} \longrightarrow s_{ij} = \left[1 - \frac{\|f_i - f_j\|_2}{\max_{k,l}(\|f_k - f_l\|_2)}\right]^2$$

- **Proposition:** Minimizing the resulting loss does encourage properties (P1) and (P2).
  But: The derivatives of this loss function are *not* continuous.

- **Solution:** A quadratic approximation with the same global minimum can alleviate this problem:

$$L_{quad} = \frac{1}{N^2} \sum_{i,j=1}^{N} ((1 - s_{ij})\Delta - D_{ij})^2$$

$$D_{ij} = \| E(x_i) - E(x_j) \|_2$$

$\Delta \in \mathbb{R}$ is a hyperparameter

Implicit Hard-Negative Mining

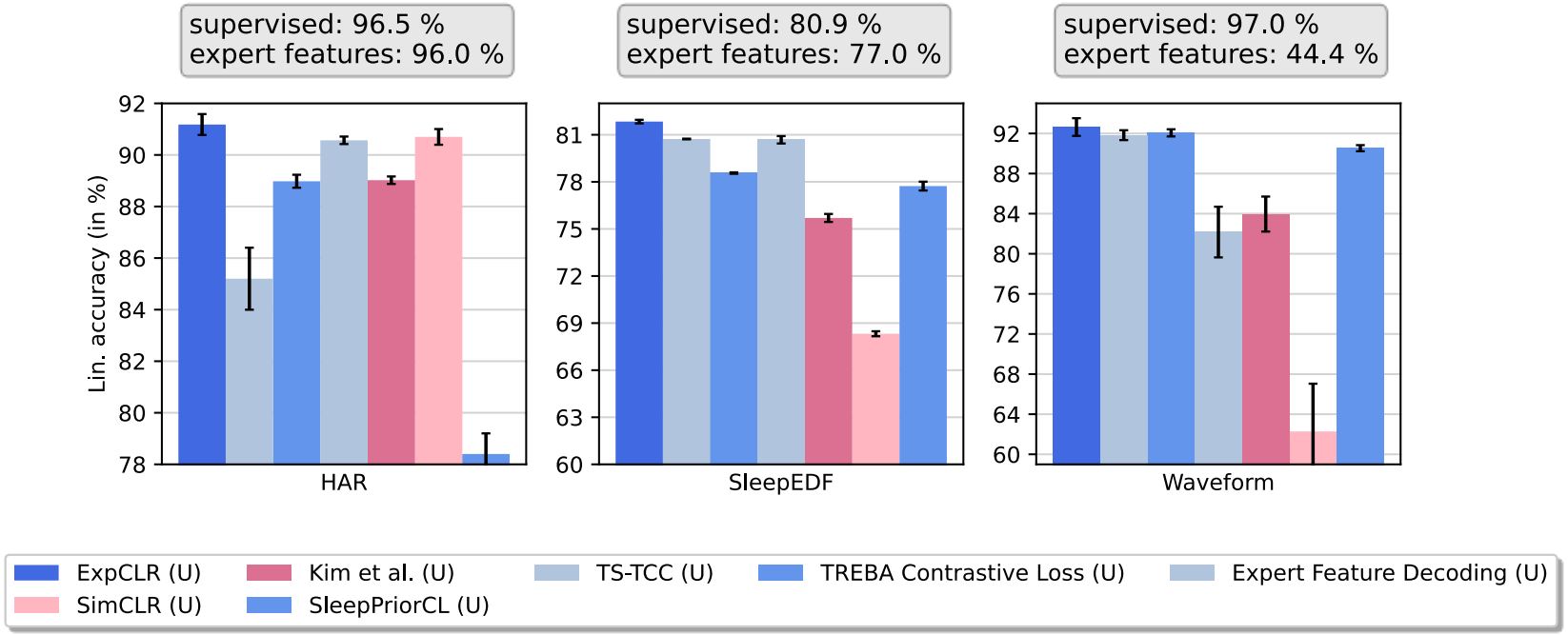
- Hard-negative mining (HNM):
  - **Explicit HNM:** Only train on pairs (i, j) with largest individual loss
  - **Implicit HNM:** Implicitly put more weight on those pairs (i, j) with larger individual loss

- Implicit HNM usually leads to superior performance and can be employed via a SoftMax-like loss function:

$$L^{\tau}_{ExpCLR} = \tau \log\left[\frac{1}{N^2}\sum_{i,j=1}^{N} \exp\left(\frac{((1-s_{ij})\Delta - D_{ij})^2}{\tau}\right)\right], \qquad \tau \in \mathbb{R} \text{ is a hyperparameter (we choose } \tau = 1)$$

Results
Unsupervised Comparison

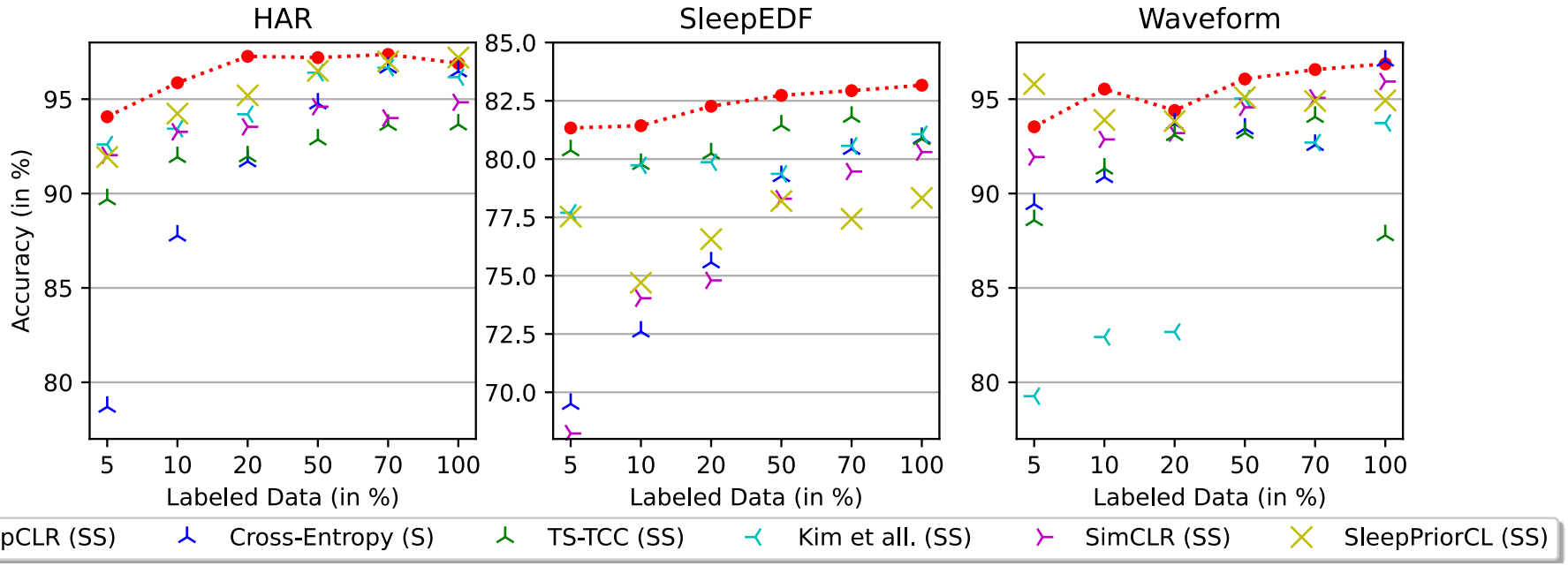
Unsupervised (U): Comparison of linear classification accuracies in the representation space, learned without any labels



- We outperform all other methods
- Surpass supervised performance on SleepEDF

Results
Semi-Supervised Comparison

Semi-Supervised (SS): Comparison of linear classification accuracies on representation space, learned on a fraction of labeled data



- We outperform all other methods
- Competing methods have drastically varying performances
- Even with 5% of labeled data we outperform some state-of-the-art-methods