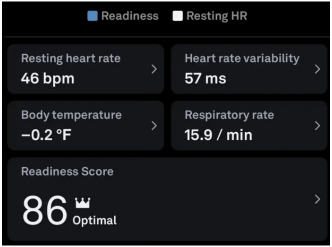
Self-Supervised Contrastive Learning on Oura Ring Data with SimCLR

This project applies SimCLR-style contrastive learning to a four-year longitudinal dataset of physiological features using a lightweight MLP encoder and a custom augmentation pipeline.

Goal: Learn embeddings that capture meaningful variation in the data without using labels.

ŌURA



🟦 Readiness □ Resting HR

Resting heart rate
46 bpm

Heart rate variability
57 ms

Body temperature
−0.2 °F

Respiratory rate
15.9 / min

Readiness Score
86 Optimal

Figure: Example of Oura daily physiological data

What is Contrastive Learning?

- Self-supervised learning method that brings similar data samples together and pushes dissimilar samples apart in latent space
- Does not require annotated labels
- Constructs positive pairs through data transformations
- Enables extraction of discriminative features from subtle patterns by learning to distinguish between augmented versions versus unrelated samples
- Works well when labeled data is sparse, or where supervised models struggle due to poor generalization under imbalance

Reference:

Chen et al. (2023). *What Makes Good Contrastive Learning on Small-Scale Wearable-based Tasks?*

SimCLR Workflow: Oura Data (Part 1)

1. **Input Data:**
   Daily z-scored physiological summaries from the Oura ring, including HRV, temperature, sleep, and activity metrics.
   → 27 features used after dropping missing values.
2. **Data Augmentation:**
   Transform sample into two correlated views:
   - Additive Gaussian noise
   - Random feature dropout (10%)

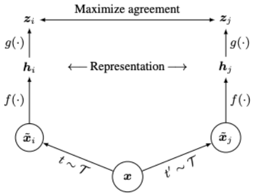


Figure: SimCLR framework (Chen et al., 2020)

SimCLR Workflow: Oura Data (Part 2)

3. **Encoder Network $f(\cdot)$:**
   A simple 2-layer MLP: `Linear(256)` → `ReLU` → `Linear(128)`
   → Outputs 128-dimensional representations.

4. **Contrastive Loss (NT-Xent):**
   Cosine similarity used to compare embeddings:
   → Positive pairs = augmented versions of the same sample
   → Negatives = all other samples in the batch Includes temperature scaling (`T = 0.1`) to tune sharpness of contrast

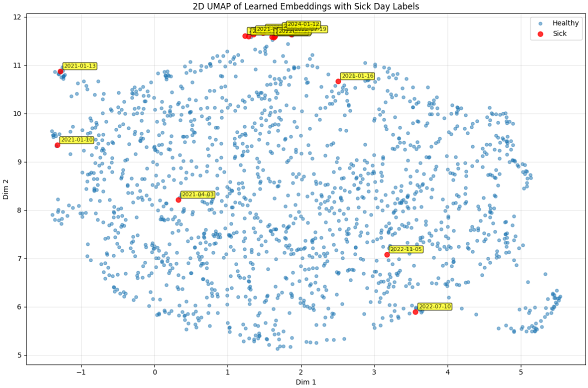
5. **Training:**
   Model trained using Adam optimizer, for 100 epochs with a batch size = 64

6. **Evaluation:**
   After training, embeddings visualized using UMAP (2D)
   → Clustering of “sick” vs “healthy” days reveals structure

Learned Embeddings Show Evidence of Health Structure

We project the 128-dimensional embeddings onto a 2D space using UMAP.

- Approx. 70% of "Sick" days (in red) tend to cluster in one distinct manifold
- "Healthy" days (in blue) span broader regions of the latent space.



Which Features Differentiate Sick Days?

Method:

- **Cohen's d effect size** between well-clustered sick days and healthy days.

Top discriminative features:

- Temp Deviation & Trend
- Total Burn & Activity Burn
- Average MET
- Equivalent Walking Distance

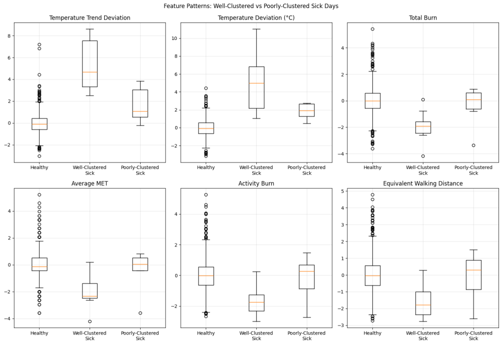


Figure: Feature pattern comparisons among cluster types

Interpretation:
Sick days that cluster well tend to have larger deviations in temperature and reduced metabolic activity.

Considerations and Future work

- **Single-subject data with imbalanced labels**
- **UMAP evaluation and stability:** Clusters are interpreted visually, and vary across seeds, but core sick clustering persists.
- **Limited augmentations:** Current perturbations mimic sensor noise, not certain of biological relevance.
- **Choice of similarity metric:** Cosine selected due to the hypothesized rhythmic nature of physiological variables. Cosine captures angular relationships, ideal for periodic signals. Also compared with Mahalanobis distance to incorporate feature variance, but struggled with matrix instability (high dimensionality).
- **Prune correlated features:** Strongly correlated inputs reduce contrastive effectiveness and can inflate similarity.
- **Early warning:** Explore whether embeddings shift *prior* to illness onset