Beyond Anomaly Detection: Trajectory Embeddings with Contrastive Learning for Predictive Maintenance*

Niklas Küchen[1,2][0009−0005−7538−0276], Liza Obermeier[1][0009−0001−5372−4024], André Ebert[1][0000−0002−0629−5722], Robert Pesch[1][0000−0003−0497−3152], and Samineh Bagheri[3][0000−0003−2488−8000]

[1] inovex GmbH, 76131 Karlsruhe, Germany
{niklas.kuechen,liza.obermeier,andre.ebert,robert.pesch}@inovex.de
[2] Hochschule Karlsruhe – University of Applied Sciences, 76133 Karlsruhe, Germany
[3] bagheri.ssa@gmail.com

Abstract. Predictive maintenance traditionally relies on static anomaly detection from isolated sensor readings. In this work, we model device health as continuous degradation trajectories through a high-dimensional reconstruction error space. Based on 20-dimensional reconstruction error vectors generated using a Temporal Convolutional Autoencoder from cleaning cycles of over 7,000 industrial kitchen appliances, we construct degradation trajectories segmented into fixed-length chunks. We compare traditional trajectory distance measures (Dynamic Time Warping, Fréchet, and Euclidean distance) against our proposed approach, HDTrajCL, a self-supervised, transformer-based TrajCL adaptation. It maps high-dimensional trajectories into fixed-dimensional embeddings in linear time, enabling fast cosine similarity comparisons over computationally expensive sequence alignments. Compressed via PCA, HDTrajCL embeddings achieve the highest spatial coherence with an AUC-ROC of 0.9728, outperforming the best algorithmic baseline. Clustering experiments validated against customer proxy labels demonstrate a homogeneity of 0.480 and an AMI of 0.379. Domain validation against rule-based defect labels confirms the learned state space localizes discrete component faults while capturing the continuous gradient of physical wear.

Keywords: Industrial Predictive Maintenance · High Dimensional Trajectory Similarity · Contrastive Learning.

* Supported by *DEXAI: Data Efficiency through eXplainable AI* - a collaborative research project funded by the Bavarian State.

1 Introduction

In Predictive Maintenance, degradation is the process of health decline of a machine over time. Mechanical and electrical components eventually wear out, requiring replacement despite regular maintenance. In commercial kitchens, unexpected outages of professional multi-function appliances cause operational disruptions and direct revenue loss. Monitoring system health is challenging due to variable daily operational usage. This work uses real-world sensor data collected from over 7.000 field-deployed appliances by our industry partner. We utilize standardized cleaning cycles as a reliable baseline for health analysis, monitored by four embedded temperature sensors across five phases. Initially, our industry partner relied on rigid threshold rules. Later, a Temporal Convolutional Autoencoder (TCAE) trained on non-defective operation was deployed as a robust alternative [1,2]. With the help of the TCAE, reconstruction error (RE) vectors are constructed based on Mean Squared Error (MSE) per sensor and phase, yielding 20-dimensional ($4 \times 5 = 20$D) vectors that quantify deviation from flawless condition. Sequential cleanings trace a continuous trajectory through this high-dimensional RE space. Comparing these trajectories requires robust sequence similarity measures. Traditional distance metrics like Dynamic Time Warping (DTW) or Fréchet distance incur quadratic complexity $\mathcal{O}(L^2)$ for trajectory length L [3,4]. Computing pairwise distance matrices for N trajectories leads to a scalability bottleneck of $\mathcal{O}(N^2)$. Deep representation learning projects variable-length data onto a fixed-dimensional continuous embedding space for direct clustering via cosine similarity. This paper addresses two central Research Questions (RQ):

1. **RQ1**: How does self-supervised deep representation learning compare to traditional distance metrics for measuring similarity between degradation trajectories - in terms of cluster purity, semantic alignment, and scalability?
2. **RQ2**: Do similar usage patterns and ageing processes manifest as distinct, traceable trajectories in the RE space?

Contributions The primary contribution is *High-Dimensional Trajectory Embeddings using Contrastive Learning* (HDTrajCL), a novel self-supervised, dual-feature attention transformer model trained via contrastive learning (InfoNCE). Adaptations from spatial trajectory learning (TrajCL [5]) include a data-driven K-Means tokenization layer for state discretization, continuous feature deltas, and operational energy features. Empirical benchmarks demonstrate the we achieve superior cluster purity and linear inference scalability with HDTrajCL.

2 Related Work and Background

Predictive maintenance shifts from rule-based diagnostics to data-driven monitoring, learning latent equipment representations from sensor data. Reconstruction methods like autoencoders and variational autoencoders train exclusively on healthy data, quantifying subsequent deviations via the RE. In an industrial study, Krasnikov et al. [2] showed unsupervised deep-learning models successfully identify anomalous behaviour in process-complex time series, proving REs meaningfully represent equipment condition. Building on this, recent prognostics research focuses on learning latent health spaces and indicators capturing gradual degradation rather than isolated fault events. However, existing approaches mostly evaluate individual observations or short windows, providing limited views of long-term degradation dynamics.

Simultaneously, trajectory analysis and self-supervised representation learning emerged as powerful tools modeling sequential data. Classical similarity measures like DTW and Fréchet distance [4] compare temporally evolving processes but require pairwise alignment with quadratic complexity. Recently, contrastive learning approaches like Contrastive Predictive Coding, TS2Vec [6], and TrajCL [5] demonstrated meaningful sequence representations are learnable from unlabeled data. Specifically, Chang et al. [5] introduced TrajCL, a dual-feature contrastive learning framework for spatial trajectory similarity. Nevertheless, existing trajectory-learning methods primarily focus on geographic movement, failing to model high-dimensional industrial degradation pathways. This work addresses this gap by treating equipment wear as trajectories through RE state space, adapting contrastive learning to non-spatial, multivariate degradation processes.

3 Data Preparation

The data pipeline transforms raw sensor logs into standardized, continuous RE trajectory chunks representing appliance wear histories.

Data Source and Preprocessing Sensor measurements are collected at 1 Hz across four primary temperature sensors (Drain Basin, Chamber Wall, Main Chamber, Boiler) during five fault-sensitive phases of automated cleaning cycles. A pretrained TCAE processes the raw sequences ($\sim 12,000$ points over ~ 200 minutes) to quantify reconstruction deviation as MSE per sensor and phase, yielding a 20D RE vector $\boldsymbol{e} \in \mathbb{R}^{20}$ per cleaning cycle [2]. Filtering over a two-year period (Nov 2023–Nov 2025) across a single hardware variant yields

0.81 million cleanings across $7,400$ appliances. To compress heavily right-skewed error distributions (10^{-6} to 10^{0}), a base-10 logarithm ($\log_{10}$) is applied. Operational stress between cleanings is captured by active energy consumption (in kWh) of the heating elements and boiler, transformed via $\ln(1+x)$. All feature dimensions are Z-score normalized and clipped at $\pm 5\sigma$.

Segmentation, Smoothing, and Trajectory Chunking To isolate uninterrupted wear cycles, device histories are partitioned at maintenance events using offline sliding window change-point detection (L_1-norm median cost, discrepancy penalty $p = 50$) [7]. Retaining only change points where the succeeding mean error decreases isolates 773 repair points across 574 appliances, yielding 8,046 independent subtrajectories. High-frequency noise is suppressed using a Savitzky-Golay polynomial filter (window length 21, polynomial order 3) [8]. Subtrajectories are segmented into overlapping fixed-window chunks bounded by $L_{\min} = 20$ and $L_{\max} = 50$ cleanings with a stride $s = 20$. This yields $n = 32,676$ overlapping chunks. Each point p_i in a trajectory chunk is formulated as $p_i = (\boldsymbol{e}_i^\top, \boldsymbol{t}_i^\top)^\top \in \mathbb{R}^{22}$, where $\boldsymbol{e}_i \in \mathbb{R}^{20}$ represents the $\log_{10}$ RE vector and $\boldsymbol{t}_i \in \mathbb{R}^2$ represents the transformed 2D energy consumption.

4 Concept

This chapter presents the conceptual approach underlying our proposed pipeline to analyse degradation trajectories. It comprises two stages, each addressing a research question from Section 1. First, HDTrajCL transforms high-dimensional, variable-length degradation trajectories into semantically meaningful, fixed-size embeddings (Section 4.1). This addresses **RQ1** in two ways: it enables comparison between different feature representations of the underlying RE vectors, and it replaces algorithmic distance measures with a similarity metric applied to the learned embeddings. Algorithmic approaches like Dynamic Time Warping (DTW) or Fréchet distance scale quadratically with trajectory length L, i.e., $\mathcal{O}(L^2)$. Building a full pairwise distance matrix for clustering adds quadratic cost in data points N, i.e., $\mathcal{O}(N^2)$. Fixed-dimensional embeddings avoid both bottlenecks. Second, domain-informed stratification separates standard cleaning cycles from heavy-duty programs that alter temperatures during rinsing and care phases, inflating REs independent of actual wear (Section 4.2). This ensures clustering reflects true degradation rather than the type of cleaning program. The resulting embeddings form the basis for downstream clustering, directly addressing **RQ2**.

4.1 Contrastive Learning for trajectory embeddings

HDTrajCL retains TrajCL's dual-feature attention design by combining spatial and structural feature embeddings [5]. For spatial features, point p_i combines 20D RE vector $\boldsymbol{e}_i$, RE delta $\Delta\boldsymbol{e}_i = \boldsymbol{e}_i - \boldsymbol{e}_{i-1} \in \mathbb{R}^{20}$ for local drift, and 2D energy consumption $\boldsymbol{t}_i \in \mathbb{R}^2$, yielding $x_{\text{spat}} = (\boldsymbol{e}_i^\top, \Delta\boldsymbol{e}_i^\top, \boldsymbol{t}_i^\top)^\top \in \mathbb{R}^{42}$. To discretize the high-dimensional RE space without grid constraints or cell explosion, HDTrajCL partitions feature space via K-Means ($k = 128$, via elbow analysis) on 5D PCA-reduced RE features. The centroids represent discrete operational wear states. During training, each RE vector maps to its nearest centroid index, linking to learnable embedding $x_{\text{struct}} \in \mathbb{R}^{d_{\text{model}}}$ optimized to capture inter-cluster relationships. During inference, this simplifies to a fast nearest-centroid lookup.

Network Architecture An overview of the HDTrajCL architecture is provided in Figure 1. Continuous spatial features x_{spat} are normalized via Layer Normalization [9] and projected to model dimension $d_{\text{model}} = 64$. Structural token embeddings $x_{\text{struct}} \in \mathbb{R}^{d_{\text{model}}}$ are additively fused with projected spatial features. Sinusoidal positional encodings [10] inject temporal order information. A 2-layer

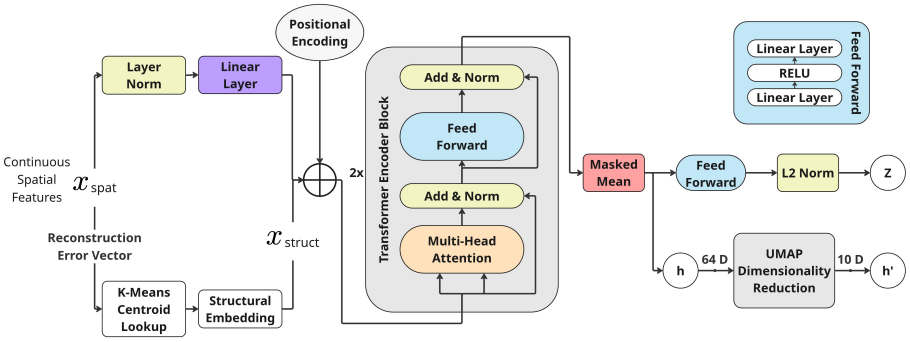


Fig. 1: Network architecture of the HDTrajCL model.

Transformer Encoder backbone ($n_{\text{heads}} = 4$, $d_{feedforward} = 256$, dropout 0.1) processes the sequence via multi-head self-attention. A boolean padding mask prevents attention to zero-padded elements in variable-length chunks. A masked mean pooling layer aggregates point-level representations into a global chunk embedding $h \in \mathbb{R}^{64}$. During contrastive training, h passes through a 2-layer MLP projection head with ReLU activations, outputting $z \in \mathbb{R}^{64}$, L2-normalized onto a unit hypersphere.

Contrastive Objective and Data Augmentations The model is trained end-to-end using the InfoNCE loss [11]:

$$\mathcal{L}\text{InfoNCE} = -\log \frac{\exp(\text{sim}(z_i, z_i^+)/\tau)}{\exp(\text{sim}(z_i, z_i^+)/\tau) + \sum k = 1^K \exp(\text{sim}(z_i, z_k^-)/\tau)} \quad (1)$$

where $\text{sim}(\cdot,\cdot)$ is cosine similarity, $\tau = 0.07$ temperature, z_i^+ an augmented anchor view, and z_k^- negative batch samples. We optimize using Adam ($\eta = 10^{-3}$, weight decay 10^{-4}, batch size 64). Augmented views are generated via four stochastic methods: 1) Gaussian Noise Addition, 2) Trajectory Truncation of sequence ends, 3) Sequence Warping via speed curve re-interpolation, and 4) Channel Masking of feature dimensions.

Intermediate Embedding Dimensionality Reduction The 64D HDTrajCL chunk embeddings h are further compressed to $h' \in R^{10}$ with different approaches, namely linear PCA, UMAP (with cosine metric), and a feed-forward Autoencoder in prior to downstream state clustering.

4.2 Domain-Specific Data Split and Downstream Clustering

Heavy-Duty Cleaning Split Initial spatial exploration revealed a dominant binary separation in embedding space. Domain analysis confirmed this split is caused by an optional heavy-duty cleaning program setting that alters cycle temperatures during rinsing and care phases (Phases 3 and 4). Because heavy-duty cleanings were under-represented in the TCAE training data, they exhibit inherently higher baseline REs. Naive clustering would partition data by cleaning program rather than by component wear. Consequently, the dataset is split into heavy-duty and non-heavy-duty subsets; clustering is executed independently on each subset and merged afterwards.

Clustering Algorithms and Composite Optimization A grid search is executed independently on both data subsets, whereby Partitioning Around Medoids (PAM) and Agglomerative Hierarchical Clustering (AC) were evaluated for precomputed distance matrices [12,13]. For learned embeddings we evaluated K-Means, Gaussian Mixture Models (GMM), K-Medoids, and AC [14,15,16,17]. The model selection itself is guided by a Composite Score $S_{\text{comp}}(c)$ balancing external validation against customer proxy labels with a simplicity penalty against over-segmentation $S_{\text{comp}}(c) = w_e \cdot S_{\text{ext}}(c) + w_s \cdot S_{\text{simp}}(k)$, where weights are set to $w_e = 0.7$ and $w_s = 0.3$. Here, $S_{\text{ext}}(c)$ is the unweighted mean of min-max normalized external validation metrics (ARI, AMI, Homogene-

ity, Completeness) computed against 12 verified customer proxy profiles (3,186 chunks across 301 devices). The linear simplicity penalty $S_{\text{simp}}(k)$ is defined as $S_{\text{simp}}(k) = 1 - \frac{k-k_{\min}}{k_{\max}-k_{\min}}$, where $k_{\min} = 3$ and $k_{\max} = 15$. Finally, Maximizing $S_{\text{comp}}(c)$ selects models that capture meaningful physical wear states while penalizing arbitrary cluster fragmentation.

5 Evaluation

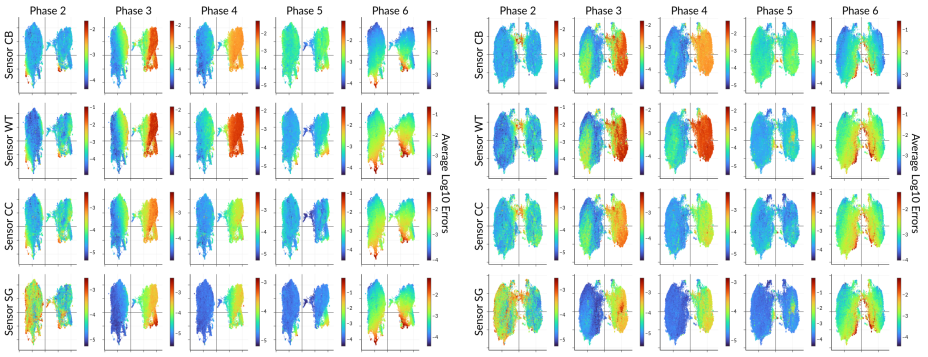
This chapter empirically evaluates the concept (Section 4) regarding the RQs in Section 1. Section 5.1 compares learned embeddings against algorithmic distance measures (**RQ1**), while Section 5.2 assesses whether the embedding space yields distinct, traceable degradation trajectories (**RQ2**). The chapter concludes with a domain validation using expert-labeled defects (Section 5.3) and an analysis of cluster transition dynamics (Section 5.4).

5.1 Comparison of Trajectory Distance Measures (RQ1)

HDTrajCL embeddings are compared against three algorithmic baselines. Euclidean distance computes point-wise distance across flattened trajectory vectors, DTW permits non-linear temporal alignment, and Fréchet distance evaluates maximum point divergence along an alignment path [3,4,18]. All three are evaluated on raw 20D RE vectors and on vectors compressed via PCA, UMAP, or an AE [19,20,21]. For HDTrajCL, dimensionality reduction is applied to the learned 64D embeddings, with similarity quantified via cosine similarity.

Qualitative Feature Projection Analysis Since the TCAE is trained only on healthy appliances (Section 1), RE magnitude quantifies deviation from flawless conditions, proxying physical wear. Representations capturing true degradation must place chunks with similar RE magnitudes spatially close. Figure 2 compares two methods. For the algorithmic baseline, pairwise DTW distances are computed first and passed to UMAP as a precomputed metric. For HDTrajCL, learned 64D embeddings are directly projected. Both projections show smooth error gradients across phases, confirming they capture physical degradation over algorithmic artifacts. Both spaces exhibit distinct "wings" separating standard from heavy-duty cleaning modes, driven by phase 3 and 4 error shifts.

Direct Spatial Coherence Evaluation Before clustering trajectories, we evaluate spatial coherence. Chunks from one subtrajectory represent continuous device wear. Under a coherent metric, subsequent overlapping chunk pairs from the same subtrajectory should exhibit lower distances than other chunks from



(a) DTW distance matrix (20D RE). **(b)** HDTrajCL embeddings.

Fig. 2: Qualitative 2D UMAP projections colored by mean RE across sensors and phases.

different ones. We quantify this via binary classification - same vs. different subtrajectory predicted from distance - reporting AUC-ROC over a random sample of $3,000$ subtrajectories ($17,272$ chunks). Best results are shown in Table 1. HDTrajCL embeddings compressed with 10D PCA achieve the highest overall score (AUC-ROC $= 0.9728$), outperforming the best algorithmic distance (DTW on 5D UMAP features at 0.9705). For algorithmic metrics, UMAP feature reduction significantly boosts DTW on raw $\log_{10}$ chunks (20D) from 0.9469 to 0.9705, acting as a non-linear noise filter. UMAP also outperforms PCA (5D, 0.9369) and AE (5D, 0.9052). DTW consistently outperforms Euclidean and Fréchet metrics.

Tab. 1: Spatial coherence evaluation via subtrajectory classification AUC-ROC.

Dist. Metric	Feature DR.	AUC-ROC↑
Euclidean	UMAP (5D)	0.9682
DTW	UMAP (5D)	**0.9705**
Fréchet	UMAP (5D)	0.9545

(a) Algorithmic Trajectory Distances

Embedding DR.	AUC-ROC↑
Raw 64D HDTrajCL	0.9707
PCA (10D)	**0.9728**
UMAP (10D)	0.9599
AE (10D)	0.9627

(b) Learned Embedding Distances

Abbreviations: Dist. = Distance, DR. = Dimensionality Reduction.

5.2 Trajectory Clustering and State Space Quantization (RQ2)

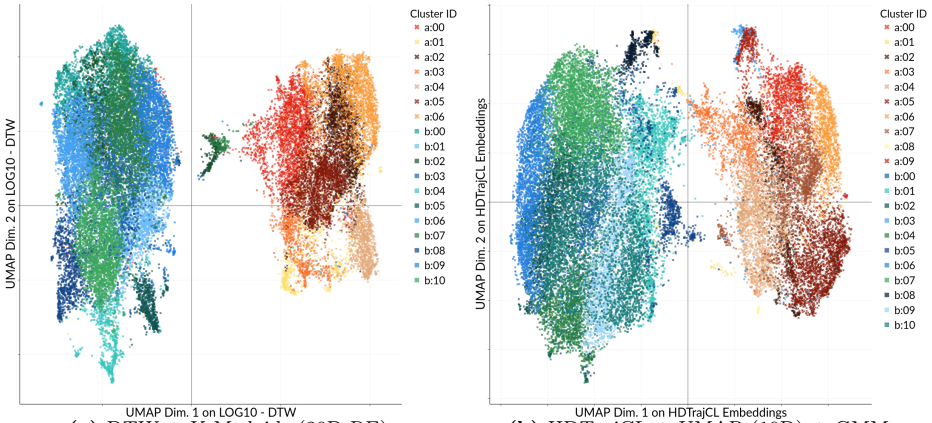
External Validation via Customer Proxy Labels Since physical wear progresses continuously without hard boundaries, internal metrics (Silhouette, Davies-Bouldin) penalize natural overlap. Evaluating semantic purity, 12 customer profiles with standardized usage ($3,186$ chunks across 301 devices) serve as external proxies. Grid search over cluster numbers $k \in [3, 15]$ uses the Composite Score. We select $k = 13$ clusters for standard cleaning and $k = 8$ for heavy-

duty subsets. Table 2 compares top-performing configurations. The embedding pipeline (HDTrajCL + UMAP 10D + GMM) yields the highest mean score (0.344), Homogeneity (0.480), and AMI (0.379). The best algorithmic baseline (Raw $\log_{10}$ + DTW + K-Medoids) attains a 0.333 mean score and higher ARI (0.269). Figure 3 displays the best cluster topologies in 2D space.

Tab. 2: Comparison of top clustering configurations evaluated against customer proxy labels.

Algorithmic Trajectory Distances								
Feat. DR	**Dist. Metric**	**Clust.**	**AMI ↑**	**ARI ↑**	**Hom. ↑**	**Comp. ↑**	**Mean ↑**	**Rank ↓**
Raw $\log_{10}$	DTW	PAM.	**0.343**	**0.269**	0.412	0.308	**0.333**	1
Raw $\log_{10}$	Euclidean	PAM	0.337	0.195	0.422	0.297	0.313	2
UMAP	Euclidean	PAM	0.336	0.204	0.413	0.296	0.312	3
PCA	DTW	PAM	0.329	0.160	**0.438**	0.280	0.302	4
Learned Embedding Distances								
Embedding DR		**Clust.**	**AMI ↑**	**ARI ↑**	**Hom. ↑**	**Comp. ↑**	**Mean ↑**	**Rank ↓**
UMAP (10D)		**GMM**	**0.379**	**0.189**	0.480	**0.326**	**0.344**	1
UMAP (10D)		K-Means	0.376	0.185	0.475	0.325	0.340	2
UMAP (10D)		K-Med.	0.369	0.186	0.471	0.319	0.336	3
PCA (10D)		GMM	0.348	0.146	**0.481**	0.290	0.316	5

Abbreviations: Feat. = Feature, DR. = Dimensionality Reduction, Dist. = Distance, Hom. = Homogeneity, Comp. = Completeness, Clust. = Clustering, K-Med. = K-Medoids.



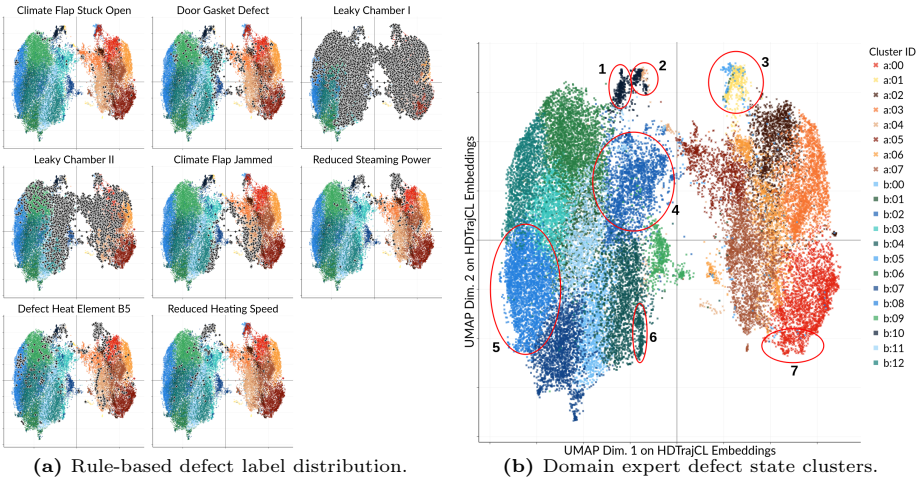
(a) DTW + K-Medoids (20D RE). **(b)** HDTrajCL + UMAP (10D) + GMM.

Fig. 3: Comparative 2D UMAP cluster partition visualizations for top algorithmic baseline vs. learned embedding pipeline.

5.3 Domain Validation with Rule-Based Defect Labels

To evaluate fault alignment, rule-based defect labels are aggregated across chunks (40% cleaning threshold). Figure 4 visualizes defect distributions alongside expert cluster verification. Discrete mechanical failures exhibit highly localized em-

bedding space patterns. A jammed climate flap causes an unexpected Chamber Wall temperature rise during Phase 6 (Drying), forming dense peripheral clusters. Reduced steaming power manifests as rapid boiler heating lag, isolating fault regions near manifold boundaries. Unlike discrete breaks, chamber leakiness from gasket degradation shows a continuous gradient from healthy outer regions (RE $\sim 10^{-6}$) toward central defect states (RE $\sim 10^{0}$). Expert inspection confirms learned clusters isolate faults like boiler defects, air leakages, and jammed flaps.



(a) Rule-based defect label distribution. **(b)** Domain expert defect state clusters.

Fig. 4: Validation of learned state space using rule-based defect labels and domain expert verification.

5.4 Cluster Transitions and Degradation Pathways

Wear-out history is modeled as a directed transition graph $G = (V, E)$, with cluster nodes V and edges E recording consecutive chunk transitions. Self-transitions (devices remaining in the same cluster) comprise 70–90% of steps and are excluded, reflecting persistence over change. Remaining transitions concentrate on few neighboring states: in the standard cleaning subset, several states funnel over 60% of outgoing transitions into two clusters. In the heavy-duty subset, three states absorb over 80% of one state's outgoing transitions. This pattern indicates directional drift over random wandering. Appliances progress through limited adjacent intermediate states, matching the continuous, gradient-like degradation observed in Section 5.3 between healthy peripheral and central defect states. These structured, predictable transitions inform an appliance's likely next states, establishing a basis for forecasting future defect probabilities.

6 Conclusion and Future Work

We introduced HDTrajCL, a self-supervised contrastive learning framework representing industrial degradation as trajectories through a high-dimensional RE space. Combining REs, temporal dynamics, and operational energy features, it learns compact embeddings capturing degradation behaviour over time. For **RQ1**, learned embeddings provide higher semantic quality and scalability than traditional trajectory similarity measures. HDTrajCL achieved the strongest performance, avoiding quadratic computational costs of alignment-based methods like DTW and Fréchet distance. Regarding **RQ2**, similar usage and ageing processes manifest as structured, traceable trajectories within the learned space. Specific customer profiles populate clusters, while rule-based defect labels and expert validation confirm the localization of distinct fault states. Appliances progress through recurring transition pathways, indicating continuous, predictable degradation patterns rather than random state changes.

Overall, these findings demonstrate trajectory representation learning's potential to advance Predictive Maintenance from static anomaly detection toward modelling equipment health evolution. Although limited to a single appliance family using REs, the learned state space captures meaningful degradation dynamics. Future work will validate these dynamics via field inspections, integrate domain-specific features, extend the approach to other appliances, and explore predictive applications like degradation forecasting and remaining useful life estimation.

References

1. Park, J., Park, Y., Kim, C.I.: TCAE: Temporal Convolutional Autoencoders for Time Series Anomaly Detection. In: 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN). pp. 421–426 (Jul 2022)
2. Krasnikov, S., Meitz, L., Bagheri, S., Heider, M., Schöler, T., Hähner, J.: Unsupervised anomaly detection in process-complex industrial time series: A real-world case study. arXiv preprint arXiv:2604.13928 (2026)
3. Berndt, D.J., Clifford, J.: Using Dynamic Time Warping to Find Patterns in Time Series. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. pp. 359–370. AAAIWS'94, AAAI Press, Seattle, WA (Jul 1994)
4. Fréchet, M.M.: Sur quelques points du calcul fonctionnel. Rendiconti del Circolo Matematico di Palermo (1884-1940) 22(1), 1–72 (Dec 1906)
5. Chang, Y., Qi, J., Liang, Y., Tanin, E.: Contrastive trajectory similarity learning with dual-feature attention. arXiv preprint arXiv:2210.05155 (2023)

6. Li, X., Zhao, K., Cong, G., Jensen, C.S., Wei, W.: Deep Representation Learning for Trajectory Similarity Computation. In: 2018 IEEE 34th International Conference on Data Engineering (ICDE). pp. 617–628. IEEE, Paris (Apr 2018)
7. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Signal Processing 167, 107299 (Feb 2020)
8. Savitzky, Abraham., Golay, M.J.E.: Smoothing and Differentiation of Data by Simplified Least Squares Procedures. Analytical Chemistry 36(8), 1627–1639 (Jul 1964)
9. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
11. van den Oord, A., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding (Jan 2019)
12. Sokal, R.R., Michener, C.D.: A Statistical Method for Evaluating Systematic Relationships. University of Kansas (1958)
13. Kaufmann, L., Rousseeuw, P.J.: Partitioning Around Medoids (Program PAM). In: Finding Groups in Data, pp. 68–125. John Wiley & Sons, Ltd (1990)
14. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, vol. 5.1, pp. 281–298. University of California Press (Jan 1967)
15. Reynolds, D.: Gaussian Mixture Models. In: Li, S.Z., Jain, A. (eds.) Encyclopedia of Biometrics, pp. 659–663. Springer US, Boston, MA (2009)
16. Ward Jr., J.H.: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58(301), 236–244 (Mar 1963)
17. Park, H.S., Jun, C.H.: A Simple and Fast Algorithm for K-medoids Clustering. Expert Systems with Applications 36(2, Part 2), 3336–3341 (Mar 2009)
18. Eiter, T., Mannila, H.: Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria (1994)
19. Bishop, C.M., Bishop, H.: Continuous Latent Variables. In: Bishop, C.M., Bishop, H. (eds.) Deep Learning: Foundations and Concepts, pp. 495–531. Springer International Publishing, Cham (2024)
20. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. Journal of Open Source Software (2018)
21. Bishop, C.M., Bishop, H.: Autoencoders. In: Bishop, C.M., Bishop, H. (eds.) Deep Learning: Foundations and Concepts, pp. 563–579. Springer International Publishing, Cham (2024)