Inferring Wind Turbine Operational State and Fatigue from High-Frequency Acceleration using Self-Supervised Learning for SCADA-free Monitoring

Yacine Bel-Hadj [1], Francisco de Nolasco Santos [1], Wout Weijtjens [1], and Christof Devriendt [1]

[1]OWI-Lab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Elsene, Belgium

Correspondence: Yacine Bel-Hadj (yacine.bel-hadj@vub.be)

Abstract. Wind-turbine operation is commonly described using Supervisory Control and Data Acquisition (SCADA) systems. ~~the vast majority of fleet-wide records available for analysis consist of 10-minute averages. These coarse aggregates obscure short transients and dynamic interactions, access is often restricted by proprietary control systems, and the data frequently contain gaps.~~

~~Wind-turbine operation is commonly described using SCADA systems.~~ While high-frequency SCADA data (e.g., 1 s resolution) exist, the vast majority of fleet-wide records available for analysis consist of ~~10-minutes aggergates~~10-minute averages. These coarse aggregates ~~make them insensitive to short transients . Additionally,~~ obscure short transients and dynamic interactions; access is often restricted by proprietary control systems~~, and the records~~ ; and the data frequently contain gaps. To address these limitations, a SCADA-free approach is developed in which operational states are inferred directly from high-frequency nacelle acceleration, a sensor that is increasingly being installed across wind farms, e.g. to monitor loads. The proposed method is based on a denoising autoencoder, to which a Domain-Adversarial Neural Network (DANN) mechanism and a Deep Embedded Clustering (DEC) self-supervision are added. Compact ~~eight-dimensional~~ six-dimensional representations of one-minute vibration spectra between 0 and 3 Hz are learned. Turbine-specific signatures are suppressed through a domain-adversarial regularization, leading to turbine-invariant embeddings that capture a generalized representation of turbine dynamics. A self-supervised DEC objective structures the ~~latent space~~ latent-space into discrete and physically meaningful operational regimes. DEC facilitates the post-hoc analysis of the learned embedding. Training is performed on data from a ~~22~~ 11 out of 44 turbines offshore wind farm sampled at 31.25 Hz, while SCADA signals are used only for validation. Strong correspondence is observed between the learned embeddings and pitch, rotor speed, power, and wind speed, with normalized mutual information above 0.8. Turbine invariance is verified through mutual-information analysis between embeddings and turbine identity. This analysis also reveals clusters within the wind farm and indicates whether the learned representation can be consistently applied across different turbines. As an auxiliary validation, regression models were trained on the learned embeddings to predict 10-minute damage-equivalent moments (DEM). The regressors were fitted using data from only five strain-instrumented turbines and then applied fleet-wide. Accurate fatigue predictions were obtained across all turbines $R^2 = 0.96$, surpassing SCADA-based baselines. This demonstrates that the learned embeddings generalize beyond operational

description and contain sufficient load-related information to support fleet-wide fatigue estimation, enabling high-resolution monitoring without dependence on SCADA.

Keywords. operational insight; deep embedded clustering; domain-adversarial neural networks; autoencoder; representation learning; wind farm; cross-turbine generalisation; damage estimation; acceleration

1 Introduction

Recent years have seen offshore wind growing into a cornerstone of Europe's renewable energy expansion, with turbines steadily increasing in size and farms being installed at greater distances from shore (Soares-Ramos et al., 2020). This development has intensified the demand for reliable monitoring of the assets, which are subject to a harsh environmental and operational loads (Weijtens et al., 2016). Ensuring the long-term safety and efficiency of these assets requires not only tracking structural integrity but also attaining a clear understanding of their dynamic behaviour under realistic operating conditions. Wind turbines are inherently time-varying systems whose responses depend on a wide range of factors, including wind speed, blade pitch angle, wind direction, and the interaction of rotating components such as rotor blades and the tower (Zhao et al., 2020). These influences give rise to a broad spectrum of operating dynamics, meaning that structural responses cannot be meaningfully interpreted without knowledge of the underlying operational state (Ozturkoglu et al., 2024). This becomes even more pertinent for modern wind farms where, due to design improvements (Byrne et al., 2019), structural reserves have been diminished and fatigue has become an operational concern. With fatigue – and therefore, how long turbines may be operated – being inextricably linked with the turbine's operational state, accurate state description has become fundamental for operators.

More broadly, when monitoring such assets, knowledge of their operational context is indispensable. It provides the basis not only for Structural Health Monitoring (SHM), but also performance analysis, fault detection, condition monitoring, and fatigue-life assessment, all of which underpin safer and more cost-effective wind energy production. The importance of operational state information is reflected in international standards. At the design stage, IEC 61400-1 (IEC, 2019) defines a catalog of Design Load Cases (DLCs) that turbines must withstand under prescribed operating and environmental scenarios. For monitoring, IEC 61400-25-6 (2016) (IEC, 2016) introduces the concept of "operational state bins": a grouping mechanism intended to ensure that signals are only compared under similar conditions. In practice, however, the proposed binning in IEC (2016) is reduced to power alone, a simplification that is far too coarse for SHM where structural dynamics are more nuanced. For example, a rotor lock and an idling turbine may produce comparable power outputs yet represent fundamentally different dynamic states. In the specific case of DEM estimation and farm-wide extrapolation, a wide range of approaches has been prescribed, from physics-guided neural networks (de N Santos et al., 2024) to probabilistic models (Hlaing et al., 2024; Avendano-Valencia et al., 2020; Singh et al., 2024). However, all studies presuppose the use of SCADA (along with acceleration, for some) to prediction fatigue loads. The SCADA-dependency is so pronounced that in (de N Santos et al., 2021), where a comparative study of model performance based on different SCADA (10-min, 1s) and accelerometer (low- and high-quality) instrumentation scenarios, an acceleration-only approach is not even equated. In (de N Santos et al., 2023), a farm-wide DEM estimation study on real data, the largest errors were traced to SCADA's insufficient resolution. Short transients were not cap-

tured, and the assumption of constant yaw angle over 10 minutes often failed. Recent studies have therefore stressed the need to annotate operating conditions to make condition monitoring results interpretable (Daems et al., 2023). The reliability of such annotation is further linked to the ability to evaluate operational conditions consistently, which has been recognized as central to the stable operation of wind farms and power grids (Chu et al., 2019).

Traditionally, operational state annotation relies on SCADA systems, where multiple variables (power, rotor speed, pitch angle, wind speed) are thresholded into categories such as operating, idling, or stopped. Alternatively, data-driven approaches have attempted to automate this process: Chu et al. (2019) used principal component analysis (PCA) to reveal dominant operational modes, while Bette et al. (2023) applied bisecting k-means clustering to SCADA correlation matrices. Yet both thresholding and clustering remain limited by SCADA itself: access is often restricted, signals may be inconsistent across manufacturers, and 10-minute averaging obscures transients such as load spikes or start–stop events (Korkos et al., 2022). In addition, SCADA annotation depends on multiple signals, so the absence of a single variable can invalidate state classification—a common issue noted by Hameed et al. (2009). In contrast, acceleration-based approaches require only a single measurement modality and offers higher temporal resolution and fewer failure points . Such signals complement, rather than replace, SCADA by enabling finer detection of operational transients.

The increasing deployment of accelerometers through IoT technologies now makes it possible to collect high-frequency vibration data across entire farms. These measurements embed signatures of both environmental forcing and structural dynamics, providing a powerful alternative to infer operational states directly from vibrations. When SCADA is unavailable or unreliable, vibration-derived annotations can fill the gap, offering insight into downtime, start–stop behaviour, and fatigue-relevant transients. Leveraging these high-frequency signals for operational inference is a promising direction.

Having established the need for SCADA-independent operational inference, the central challenge is to extract operational states directly from high-frequency vibration data without labeled examples. This requires identifying the essential structure within rich, high-dimensional measurements while maintaining their physical interpretability. Representation learning provides a natural framework for this task. Autoencoders (AEs) (Hinton and Salakhutdinov, 2006) and other deep representation-learning methods (LeCun et al., 2015; Bengio et al., 2013) learn compact ~~latent spaces~~ latent-spaces that capture dominant patterns of variation while suppressing noise and incidental detail. When applied to physical sensor data, such embeddings often acquire semantic meaning that reflects the underlying system dynamics rather than the raw signal characteristics (Ranzato et al., 2012; Vincent et al., 2010; de Nolasco Santos et al., 2025; Bel-Hadj et al., 2022, 2025; Bel-Hadj and Weijtjens, 2022). Modern AEs extend this principle by incorporating design objectives that encourage disentanglement, hierarchical organization, and clusterability (Tschannen et al., 2018). These inductive properties, often referred to as meta-priors (Bengio et al., 2013), are particularly valuable in vibration-based monitoring where a limited number of physical processes such as loading, resonance, and rotor interaction govern the measured response. At the core of these extensions lies the intrinsic meta-prior of the autoencoder itself, which assumes that data can be efficiently represented through a lower-dimensional encoding that preserves the information required for reconstruction. In other words, the AE implicitly promotes representations that compress the signal while retaining its functional structure. Building on these principles, the present work introduces two additional priors tailored to wind-turbine monitoring: a domain-adversarial regularization that enforces turbine-invariant embeddings, and

a clustering objective that structures the ~~latent space~~ latent-space into compact and interpretable operational regimes. These ideas have recently been applied within SHM. For example, convolutional autoencoders have been used to distinguish train directions and axle counts from bridge measurements in an unsupervised setting (Bel-Hadj et al., 2022). Denoising variants improve robustness by reconstructing clean inputs from corrupted observations, which encourages embeddings that generalize across operating conditions (Vincent, 2011). Although contrastive self-supervised methods have also shown promise (Liu et al., 2021; Rahimi Taghanaki et al., 2023), autoencoders remain a simple, and effective for unsupervised operational-state inference in large-scale structural monitoring.

While autoencoder frameworks provide a means to derive compact and informative embeddings, such representations often retain individual turbine biases when transferred across different assets. In wind farms, for example, turbines exhibit subtle yet systematic variations in resonance, foundation stiffness, or sensor placement, which can be encoded in the ~~latent space~~latent-space. This challenge is central to the emerging field of Population-Based Structural Health Monitoring (PBSHM) (Bull et al., 2020), where the objective is to transfer knowledge across a fleet of nominally identical structures while accounting for their inherent variability. In this context, the encoder–decoder can be interpreted as learning a population *form* (a unified functional representation that captures the essential operational dynamics shared across turbines while tolerating structured variability between them). Such a form provides a common reference against which future measurements can be assessed, enabling consistent operational inference across the fleet. One prominent solution to having a unified functional representation: is domain-adversarial learning, which explicitly enforces invariance to domain differences. The domain-adversarial neural network (DANN) (Ajakan et al., 2014) extends the adversarial training paradigm of Generative Adversarial Networks (GANs) to representation learning by coupling the main task with a domain classifier connected through a gradient reversal layer. This forces the encoder to produce embeddings that are expressive for the main task while remaining indistinguishable across domains (i.e., different turbines). Building on this principle, recent studies have demonstrated the versatility of DANN in vibration-based monitoring: Mao et al. (2020) achieved improved transfer performance in bearing fault diagnosis under variable working conditions with a structured DANN, Li et al. (2025) proposed a partial conditional adversarial network to transfer damage knowledge from numerical models to full-scale structures, and Li et al. (2023) applied DANN to bridge monitoring by aligning finite element simulations with field data. Similarly, Martakis et al. (2023) fused domain adaptation with feature engineering to classify unseen damage states in shake-table tests of real buildings. Collectively, these applications underscore the potential of adversarial domain adaptation for mitigating domain shifts in SHM tasks. However, its application to operational state inference in wind turbines—where turbine-specific biases are particularly pronounced—remains unexplored. Beyond adversarial approaches such as DANN, PBSHM has also explored alternative alignment strategies such as balanced distribution adaptation (BDA) (Gardner et al., 2022), although these methods are typically applied to the transfer of diagnostic knowledge, whereas our focus is solely on learning domain-invariant embeddings without transferring damage labels.

Complementing the DANN regularization, we incorporate *Deep Embedded Clustering* (DEC) (Xie et al., 2016), a self-supervised framework that jointly learns feature representations and cluster assignments, thereby structuring the ~~latent space~~ latent-space into compact and interpretable regions and facilitating post-hoc analysis of the learned embedding. DEC has proven effective in other domains—for instance, convolutional autoencoders coupled with DEC have been used to separate

vibroseismic, highway-traffic, and airport-noise sources (Snover, 2020). To the best of the author knowledge, DEC and its derivatives have ~~not yet been applied to~~ seen limited application in SHM and wind-turbine monitoring.

Together, DANN and DEC act as complementary inductive priors on the ~~latent space~~latent-space: DANN enforces turbine-invariant representations, while DEC promotes clusterability and interpretability aligned with physical operating regimes.

Motivated by these developments, we ask: Can wind turbine operational state be inferred directly from high-frequency acceleration, without relying on SCADA during training? We investigate this question on a 44-turbine offshore wind farm, using acceleration sampled at 31.25 Hz. Our approach learns compact ~~eight-dimensional~~ six-dimensional latent embeddings from one-minute spectrograms via a domain-adversarial autoencoder that enforces turbine invariance while preserving operational structure and DEC to facilitate the ~~interpretabliity of the latent dimension and force~~ interpretablity of the latent-space and promotes its clusterability.

Contributions. This work advances wind-turbine monitoring by: (i) introducing an **acceleration-only operational-state inference** framework that learns compact latent representations directly from vibration spectrograms; (ii) achieving **cross-turbine generalization** through domain-adversarial training, enabling fleet-wide deployment without per-turbine retraining; (iii) integrating **Deep Embedded Clustering (DEC)** within the autoencoder to jointly learn turbine-invariant and discretized ~~latent spaces~~latent-spaces, yielding interpretable representations aligned with distinct operational regimes; and (iv) demonstrating **practical utility** through damage-equivalent moment estimation, illustrating how the learned embeddings support structural-health monitoring and fatigue assessment.

2 Materials and Methods

This section is organized as follows. First, the offshore wind-farm dataset and its instrumentation are described to establish the sensing basis of the study. Next, the preprocessing applied to the raw acceleration data is outlined. The representation-learning framework is then introduced: vibration spectra are encoded through a denoising autoencoder whose ~~latent space~~ latent-space is jointly structured and discretized through Deep Embedded Clustering (DEC), while turbine-specific effects are suppressed via domain-adversarial regularization. This integrated architecture produces turbine-invariant, clusterable embeddings that correspond to distinct operational regimes. An auxiliary procedure for estimating 10-minute Damage-Equivalent Moments (DEM) from sequences of embeddings is also presented. Finally, the evaluation protocol is detailed, employing information-theoretic metrics to assess turbine invariance and operational informativeness.

2.1 Site Instrumentation and Operational Variability

The study is based on operational data collected from an offshore wind farm comprising 44 monopile-supported turbines that are broadly similar in structural dynamics. As noted by Bull et al. (2020), such a fleet can be treated as a homogeneous population, though minor variability in resonance frequencies arises from differences in seabed depth, fabrication, and installation tolerances. The layout and sensing configuration are shown in Fig. 1. All turbines are equipped with nacelle-mounted dedicated accelerometers that provide the high-frequency vibration data used in this study. Each nacelle unit contains $C = 3$ channels

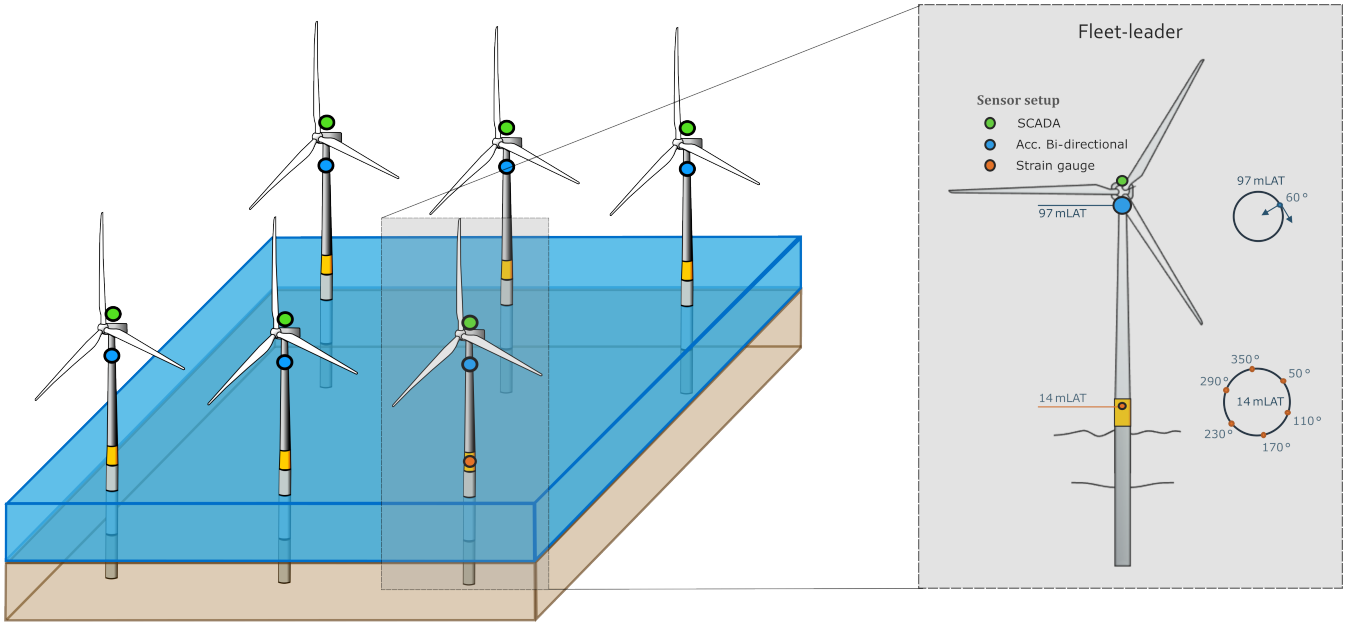

Figure 1. Schematic of the offshore wind farm and sensing layout: nacelle accelerometers (blue) provide high-frequency vibration data used for learning operational embeddings, and SCADA signals (green) provide supervisory and control measurements used only for evaluation and interpretation. Tower/monopile strain gauges (orange) are installed on a small subset of turbines - so-called fleet-leaders.

(fore–aft, side–side and vertical directions) sampled at $31.25Hz$. SCADA signals, by contrast, are recorded by the turbine control system at a low frequency of $\frac{1}{600}Hz$ (10-minute averages) and are used solely for evaluation and interpretation. Strain gauges installed near the tower–transition piece interface on five "fleet-leader" turbines are used to provide fatigue reference data but are costly; consequently, only a limited subset is instrumented, as is common in offshore monitoring (Weijtens et al., 2016). Farm-wide fatigue is typically extrapolated from these leaders using SCADA-based models (de N Santos et al., 2021). In this study, the strain-gauge measurements will be utilized only in Section 3.4 as the source of ground truth for 10-minute Damage Equivalent Moments (DEM).

2.1.1 Operational Variability

Wind turbine operation is traditionally classified from SCADA data using rule-based thresholds applied to variables such as rotor speed, blade pitch, power output, wind speed, and occasionally yaw. Typical operational states include:

- Parked/rotor lock: rotor stopped [locked], no power production.
- Ramp-down/Ramp-up: controlled deceleration/acceleration of the rotor speed.
- Idling/spinning: low rotor speed with negligible power.
- Sub-rated generation: below-rated operation with increasing power and rotor speed.

- Near/rated generation: high power production close to rated conditions.
- Curtailed/derated: power limited by control actions or high-wind derating.
- High-wind storm control: reduced power with large pitch angles to limit loads.
- Emergency stop/trip: abrupt shutdown due to protection triggers.

Such SCADA-based classification requires expert-defined thresholds; for example, distinguishing parked from idling often involves checking both rotor speed and wind speed against predefined limits. Such schemes assume stationarity, *i.e.*, that conditions remain constant over the 10-minute window. While often reasonable, this assumption hides short-term dynamics such as rotor stops, restarts. Figure 2 illustrates this point. The spectrogram of nacelle acceleration, obtained with ~~60~~65 s windows, 30 s overlap, reveals clear differences between idling and stops, as well as short-lived transitions that would not be visible in SCADA records. Restricting the spectrum to the 0–3 Hz band focuses on the dominant rotor dynamics. These patterns indicate that the 10-minute stationarity assumption does not always hold.

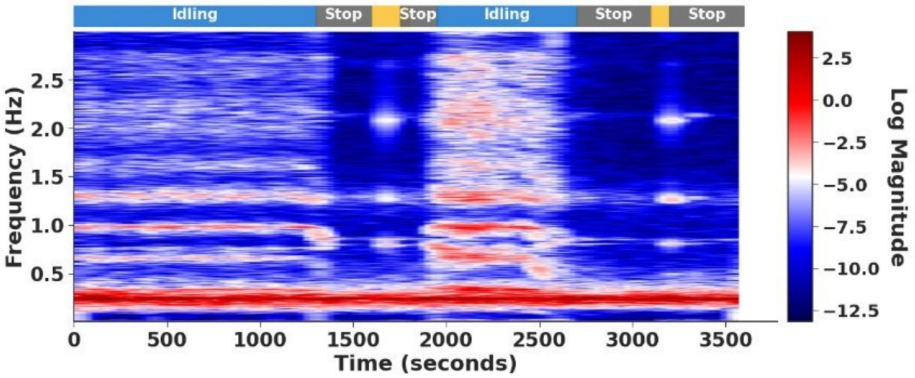

Figure 2. Log-amplitude spectrogram (0–3 Hz) of turbine acceleration with state sequence inferred from vibrations.

The acceleration-based approach developed here addresses these shortcomings. By operating directly on high-frequency vibration signals, it enables inference of operational states and transient events at sub–10-minute resolution, without the need for threshold specification. This enables finer temporal resolution of state estimation and allow event counting, complementing rather than replacing SCADA. In this study, SCADA signals are used solely for interpretation and validation of the acceleration-derived representations, not for training or direct state inference.

2.2 Representation Learning Model

The objective of this work ~~,~~is to derive compact, expressive and turbine-invariant descriptors of the acceleration signals that capture operational variability across the fleet. Such descriptors are generally referred to as *representations*, and when expressed as numerical vectors produced by a neural network, they are ~~refereed~~ referred to as *embeddings*. An embedding can

be understood as a vectorized representation of a signal–a compressed summary of an input window that preserves the essential dynamical information while discarding redundancies. Conceptually, embeddings play a similar role to manually engineered statistical features (e.g., minimum, maximum, variance), but are learned automatically by the network in a data-driven manner.

In the resulting embedding space, signals recorded under similar operational and environmental conditions are expected to map close together, while signals reflecting different dynamics should be located further apart. The structure of this space should yield well-separated clusters, whereas subtler variations (e.g., between adjacent load levels) should appear closer. To ensure that the embeddings remain physically meaningful, they are expected to exhibit strong mutual information with key supervisory variables such as rotor speed, wind speed, and blade pitch angle, the latter being particularly important as it directly defines the turbine's control state.

The dataset is composed of accelerometer measurements recorded in multiple directions (e.g., fore–aft, side–side, vertical). These signals can be ingested by the model in several ways: (i) a separate model may be trained for each direction, (ii) a shared architecture may be used while fitting independent model instances per direction, or (iii) a multi-channel architecture may be adopted in which all directions are processed jointly.

In this work, the third strategy is adopted, with each direction treated as an input channel, analogous to the color channels in image processing. The detailed multi-channel architecture is provided in Section 2.5. For clarity, the preprocessing pipeline is first described in the uni-variate (single-channel) case, and its extension to the three-channel setting is trivial.

2.2.1 Preprocessing of acceleration data

Acceleration records are segmented into 1-minute windows with a 30-second hop size, corresponding to a 50% overlap. This duration is sufficient to capture the dominant low-frequency turbine dynamics while remaining short enough to assume approximate stationarity of the signal. Formally, let the raw acceleration signal be

$$\mathbf{a} = [a_1, a_2, \ldots, a_T], \tag{1}$$

from which overlapping windows of length L and hop size H are extracted. The i-th window is denoted by

$$\mathbf{a^{(i)}} = [a_{t_i}, \ldots, a_{t_i+L-1}], \quad t_i = 1 + (i-1)H. \tag{2}$$

To prepare the time-series data for neural network input, each window of acceleration measurements is transformed into the frequency domain to capture dominant operational dynamics. A Hann window $\mathbf{w}$ is applied to reduce spectral leakage, followed by a Fast Fourier Transform (FFT) (Cooley and Tukey, 1965). The log-amplitude spectrum is then computed and truncated to the 0–3 Hz band, which covers the range of interest for tower and rotor dynamics. Only the magnitude is retained, as phase information is typically less informative in this context. The transformation is defined in Equation 3:

$$\Phi(\mathbf{a}^{(i)}) = \log\left(\left|\mathrm{FFT}(\mathbf{w} \odot \mathbf{a}^{(i)})\right| + \varepsilon\right)_{[0,3\,\mathrm{Hz}]}, \tag{3}$$

where $\mathbf{a}^{(i)}$ denotes the i-th signal window, $\mathbf{w}$ is the Hann window, and ε is a small constant ensuring numerical stability of the logarithm. With a sampling rate of $f_s = 31.25$ Hz and window length $N = 2048$, this procedure yields approximately $F \approx 200$ frequency bins per channel.

Before being fed into the neural network, the spectra $\Phi(\mathbf{a}^{(i)})$ are scaled using min–max normalization. To avoid distortion by outliers, scaling is based on the 0.1th and 99.9th percentiles of the training distribution, computed element-wise across frequency bins. Denoting these percentiles by $q_{0.1}, q_{99.9}$, the normalized input is

$$\mathbf{x} = \widetilde{\Phi}(\mathbf{a}^{(i)}) = \frac{\Phi(\mathbf{a}^{(i)}) - q_{0.1}}{q_{99.9} - q_{0.1}}, \tag{4}$$

which maps the bulk of the data approximately into the $[0, 1]$ interval while preserving contrast in the presence of occasional extreme values.

2.2.2 Autoencoder Learning and Domain-Adversarial Training

We assume that each high-dimensional spectrum $\mathbf{x} \in \mathbb{R}^M$ (hundreds of frequency coefficients) is governed by a much smaller set of latent factors ~~$\mathbf{z} \in \mathbb{R}^L$ with $L \ll F$~~ $\mathbf{z} \in \mathbb{R}^d$ with $d \ll M$. While vibration spectra may appear complex, their variability is largely explained by a handful of physical drivers such as turbine load, control settings, and environmental conditions. For instance, increasing load raises the overall vibration energy, while rotor speed introduces harmonics at multiples of the blade-passing frequency (3p, 6p, etc.). Our objective is therefore to learn a mapping,

$$f_{\text{enc}} : \mathbf{x} \mapsto \mathbf{z} \, ,$$

such that $\mathbf{z}$ captures the salient operational patterns in a compact form.

2.2.3 Autoencoder formulation.

Autoencoders provide a natural framework for this task. A standard autoencoder consists of an encoder, f_{enc}, that compresses an input spectrum into a latent embedding $\mathbf{z}$, and a decoder, f_{dec}, that attempts to reconstruct the original signal:

$$\mathbf{z} = f_{\text{enc}}(\mathbf{x}; \theta_{\text{enc}}) \, , \qquad \hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{z}; \theta_{\text{dec}}). \tag{5}$$

Here, $\mathbf{x}$ denotes the input spectrum $\mathbf{x}$, $\hat{\mathbf{x}}$ is the reconstruction, and $\theta_{\text{enc}}, \theta_{\text{dec}}$ are the trainable parameters (weights and biases) of the encoder and decoder, respectively. The latent vector $\mathbf{z} \in \mathbb{R}^d$ (with $d \ll M$ when $\mathbf{x} \in \mathbb{R}^M$) provides the compact embedding used in downstream analysis. The reconstruction is trained by minimizing the mean squared error (MSE) between the input and the output,

$$\mathcal{L}_{\text{AE}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2. \tag{6}$$

2.2.4 Denoising criterion.

To improve robustness, we adopt the *denoising autoencoder* (Vincent, 2011), in which inputs are corrupted by additive Gaussian noise,

$$\tilde{\mathbf{x}} = \mathbf{x} + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I). \tag{7}$$

Here, the corruption ϵ represents synthetic perturbations, and its scale σ controls their strength. Choosing σ on the order of natural measurement noise encourages the model to focus on the meaningful structure of the spectra while ignoring irrelevant fluctuations. The encoder receives $\tilde{\mathbf{x}}$, while the decoder is trained to recover the clean $\mathbf{x}$.

This inductive bias can be interpreted as a restoring mechanism: when noise perturbs the spectrum away from regions of physically plausible turbine data, the model learns to pull it back. In the small-noise limit, the reconstruction function approximates the *score function* $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ (Vincent, 2011), which always points in the direction where the likelihood of real data increases most steeply. Estimating this score is important because it provides the model with a way to distinguish between meaningful operational patterns and incidental deviations. In practice, the network learns to suppress sensor noise or spurious fluctuations while retaining the stable vibration signatures that reflect turbine dynamics.

2.2.5 Domain-adversarial regularization.

While the denoising criterion ensures robustness, embeddings can still encode turbine-specific signatures (e.g. resonance frequencies or sensor placement). Such features would hinder generalization to unseen turbines and complicate the interpretation of the embedding. To address this, we employ a *domain-adversarial mechanism* (Ganin and Lempitsky, 2015), where the domain corresponds to turbine identity. This can be interpreted as a *turbine-adversarial mechanism*, whose objective is to remove turbine-specific information from the embeddings.

In practice, a domain classifier f_{dom} is attached to the encoder through a Gradient Reversal Layer (GRL). For each embedding $\mathbf{z}_i$, the classifier—implemented as a small neural network ending with a softmax layer—predicts the turbine of origin:

$$\hat{d}_i = f_{\mathrm{dom}}(\mathrm{GRL}(\mathbf{z}_i); \theta_{\mathrm{dom}}), \tag{8}$$

where $\hat{d}_i$ is the predicted turbine label, θ_{dom} are the classifier parameters, and $\mathrm{GRL}(\mathbf{z}_i) = \mathbf{z}_i$ in the forward pass but reverses the gradient during backpropagation, $\frac{\partial\, \mathrm{GRL}(\mathbf{z}_i)}{\partial \mathbf{z}_i} = -\gamma I$.

The domain loss is defined as the cross-entropy between predicted and true turbine labels:

$$\mathcal{L}_{\mathrm{dom}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{1}[d_i = k] \log p_{\theta_{\mathrm{dom}}}(d_i = k \mid \mathbf{z}_i), \tag{9}$$

where $p_{\theta_{\mathrm{dom}}}(d_i = k \mid \mathbf{z}_i)$ is the predicted probability that embedding $\mathbf{z}_i$ originates from turbine k, d_i is the true turbine identity, and N is the minibatch size. During optimization, the classifier parameters are updated to *minimize* this loss, while the encoder receives the reversed gradient and thus learns to *maximize* it—encouraging domain invariance. This adversarial interaction ensures that the latent embeddings remain informative of operational dynamics while discarding turbine-specific biases.

2.2.6 Deep Embedded Clustering (DEC)

While the denoising and adversarial objectives produce embeddings that are robust and turbine-invariant, the ~~latent space~~ latent-space remains continuous, making it difficult to interpret in terms of discrete operational modes. To reveal such regimes, we adopt the *Deep Embedded Clustering* (DEC) formulation (Xie et al., 2016), which jointly refines the encoder and a set of cluster centroids so that embeddings belonging to similar operating conditions are pulled closer together while those representing distinct dynamics are pushed apart.

The underlying idea is that the model should first form *compact clusters*—bringing together latent points that correspond to consistent vibration patterns—and then *separate* these clusters sufficiently to produce interpretable operational regimes. To achieve this balance, DEC avoids hard assignments (which can lead to unstable optimization) and instead relies on *soft associations* that gradually sharpen over time.

For each embedding $\mathbf{z}_i \in \mathbb{R}^d$, its similarity to each cluster centroid μ_j is measured using a Student-t kernel:

$$q_{ij} = \frac{\left(1 + \|\mathbf{z}_i - \mu_j\|^2/\alpha\right)^{-(\alpha+1)/2}}{\sum_{j'} \left(1 + \|\mathbf{z}_i - \mu_{j'}\|^2/\alpha\right)^{-(\alpha+1)/2}}, \tag{10}$$

where q_{ij} denotes the *soft assignment* probability of sample i to cluster j. Following Xie et al. (2016), α is set to 1 so that kernel has a heavy tail, ensuring that not only nearby point are attracted to the cluster center, which stabilizes cluster formation. The heavy-tailed kernel ensures that nearby points contribute strongly while distant ones exert diminishing influence, promoting smooth cluster boundaries.

To make clusters progressively more distinct, DEC defines a sharpened *target distribution*:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_{j'} (q_{ij'}^2 / \sum_i q_{ij'})}, \tag{11}$$

which amplifies confident assignments (large q_{ij}) and down-weights uncertain ones. Intuitively, q_{ij} expresses how much a point currently belongs to a cluster, while p_{ij} represents where it *should* belong as training refines the latent structure. For instance, consider a sample located between two neighboring regimes: if its current soft assignments are $q_{i1} = 0.6$ and $q_{i2} = 0.4$, the target distribution will become $p_{i1} \approx 0.69$ and $p_{i2} \approx 0.31$ after sharpening. This numerical shift increases the weight of the more confident cluster, gently pulling the sample toward centroid 1. As training proceeds, each embedding migrates toward its most representative cluster.

The clustering loss minimizes the Kullback–Leibler divergence between the two distributions:

$$\mathcal{L}_{\text{DEC}} = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{12}$$

thereby encouraging embeddings to move closer to their respective centroids. Each centroid acts as a gravitational attractor in the ~~latent space~~latent-space, continuously pulling nearby embeddings toward a compact configuration and enhancing separation between clusters.

In practice, DEC training proceeds in two stages. First, the encoder is pretrained to solely reconstruct the input to obtain a stable and physically meaningful representation. Then, resulting embeddings are clustered using k-means to initialize the

centroids μ_j. In the second stage, the DEC objective is introduced and jointly optimized along with initial reconstruction task, gradually organizing the ~~latent space~~ latent-space into discrete, interpretable regions that correspond to turbine operating regimes.

2.2.7 Combined objective and training schedule.

The encoder–decoder ~~system is optimized under a composite loss that integrates~~ is trained using a composite objective that balances reconstruction fidelity, turbine invariance, and cluster compactness:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{dom}\text{domain}} + \beta \mathcal{L}_{\text{DEC}}, \tag{13}$$

where $\mathcal{L}_{\text{rec}}$ is the mean-squared reconstruction error ~~between clean inputs and reconstructions~~ (Eq. 6), ~~$\mathcal{L}_{\text{dom}}$~~ $\mathcal{L}_{\text{domain}}$ the domain-adversarial cross-entropy ~~loss~~ (Eq. 9), and $\mathcal{L}_{\text{DEC}}$ the ~~clustering regularization term~~ Deep Embedded Clustering regularizer (Eq. 12). The ~~coefficients~~ weights λ and β are epoch-dependent ~~weights that are gradually increased according to a staged schedule: (i) pretrain the denoising autoencoder for $t_{\text{warm}} = 100$ epochs using only~~ and are chosen heuristically based on the observed training dynamics rather than from a principled optimum.

Training follows a staged schedule. First, the autoencoder is trained using $\mathcal{L}_{\text{rec}}$ ~~, (ii) progressively activate the adversarial regularizer to~~ alone until epoch $t_{\text{start,dann}}$, allowing the encoder–decoder pair to learn a stable and physically meaningful reconstruction manifold.

Second, the domain-adversarial objective is activated and the weighting coefficient λ is increased linearly from 0 to $\lambda_{\max}$ over the next $t_{\text{duration,dann}}$ epochs. The gradient reversal mechanism forces the encoder to suppress turbine-specific ~~signatures, and (iii) introduce the DEC term~~ information; in principle, the domain-classification loss $\mathcal{L}_{\text{domain}}$ should rise toward the random-guess baseline as turbine identity becomes unrecoverable from the embeddings.

Third, after centroid initialization ~~to discretize the latent space into compact regimes. This sequencing avoids competition between objectives and prevents premature collapse of the latent manifold.~~ by k-means (with $n_{\text{clusters}} = 5$), the clustering objective is introduced at epoch $t_{\text{start,dec}}$ and β is increased linearly from 0 to $\beta_{\max}$ over $t_{\text{duration,dec}}$ epochs. The activation of $\mathcal{L}_{\text{DEC}}$ reshapes the latent-space to promote compact and separable regimes, which can temporarily increase reconstruction error as the latent geometry reorganizes; the decoder subsequently adapts and reconstruction error recovers.

This staged optimization avoids gradient interference between objectives that impose conflicting constraints on the latent-space. Reconstruction first establishes a physically grounded representation. Domain-adversarial regularization then removes turbine-specific bias without collapsing this structure. Clustering is applied last to discretize an already stable embedding. In this order, each objective refines an existing representation rather than competing to define it, which improves training stability and preserves interpretability. The resulting training dynamics are discussed in Section 3.2.

2.3 Operational regime identification from embeddings

After training, each embedding is associated with a set of soft assignment probabilities q_{ij} reflecting its similarity to the learned centroids μ_j (Eq. 10). The most probable centroid is interpreted as the current operational regime.

As a result of the combined objective, the ~~latent space~~ latent-space remains compact, turbine-invariant, and discretized into regimes that are directly interpretable in terms of turbine operation (e.g. idling, sub-rated, rated, or curtailed states).

2.4 Temporal aggregation and Damage-Equivalent Moment (DEM) inference

Although the encoder and clustering components operate on short, quasi-stationary spectral segments, fatigue-related quantities such as the 10-minute Damage-Equivalent Moment (DEM) depend on how operating conditions evolve over time. To capture these temporal dependencies, the sequence of latent embeddings produced by the encoder $\{\mathbf{z}_t\}$ produced by the encoder is processed by a recurrent model that integrates information across successive windows. In practice, a two-layer Long Short-Term Memory (LSTM) network aggregates the embeddings within each 10-minute interval and outputs a compact hidden representation summarizing the latent trajectory of the turbine's dynamic state. A linear regression head then maps this representation to the corresponding DEM value, trained under a mean-squared-error objective using reference strain-gauge measurements from the fleet-leader turbines. During this stage, the encoder parameters are frozen so that the recurrent model learns to interpret the latent dynamics rather than to modify their structure.

Here, the *linear regression head* refers to a single fully connected layer that takes as input the LSTM output representation (context vector). The LSTM and this linear layer are trained end-to-end for DEM prediction, while the encoder remains fixed.

This design introduces a clear hierarchy: the autoencoder acts as a spatial compressor that distills high-dimensional vibration spectra into a compact, turbine-invariant representation; the recurrent module integrates these representations temporally; and the regression head translates the aggregated latent dynamics into a physically meaningful fatigue indicator. Conceptually, this mirrors the structure of *world models* proposed by Ha and Schmidhuber (Ha and Schmidhuber, 2018), in which a variational autoencoder encodes raw observations, a recurrent model captures temporal evolution in ~~latent space~~latent-space, and a lightweight head operates upon that representation. In a similar spirit, the present framework constructs a latent "world view" of turbine dynamics: one that encapsulates both the instantaneous and evolving behavior of the structure-thereby enabling fatigue estimation directly from vibration-derived embeddings without recourse to SCADA data.

2.5 Implementation details: multi-branch MLP over spectra

Acceleration data are stored in one-hour files, each containing three directional components, hereafter referred to as channels. Corresponding SCADA and fatigue-related data are maintained in a database with a temporal resolution of ten minutes. For model training, the acceleration signals are segmented and transformed into spectrograms. Each 1 min spectrogram window is represented as a tensor $x \in \mathbb{R}^{B\times C\times F\times T}$, with batch size B, channels $C{=}3$ (fore–aft, side–side, vertical), frequency bins $F \approx 200$ covering 0–3 Hz, and T time frames within the minute. The network comprises per–channel encoders, a latent fusion block, and per–channel decoders, with an LSTM head used only for DEM estimation.

Per–channel encoders. For each channel $c \in \{1,\dots,C\}$, the slice $x^{(c)} \in \mathbb{R}^{B\times F\times T}$ is reshaped to (BT, F) and passed through a small ~~MLP~~ Multi Layer Perceptron (MLP) $\phi_c : \mathbb{R}^F \to \mathbb{R}^{d_c}$ (three 128–unit layers with normalization and ReLU). ~~Meaning that each~~ Each timestamp is treated as an ~~independant~~ independent sample We set $d_c{=}16$. The resulting per–frame

latents $\{z_t^{(c)}\}_{t=1}^{T}$ are concatenated:

$$z_t^{\text{cat}} = \left[z_t^{(1)}; \ldots; z_t^{(C)}\right] \in \mathbb{R}^{Cd_c}.$$

Fusion to shared embedding. A compact fusion MLP $\psi : \mathbb{R}^{Cd_c} \to \mathbb{R}^{128} \to \mathbb{R}^{d}$ (linear–norm–ReLU–linear) maps z_t^{cat} to a shared latent $z_t \in \mathbb{R}^d$. Stacking over time yields $Z \in \mathbb{R}^{B\times T\times d}$ with ~~$d{=}8$~~ $d{=}6$ used throughout.

Per–channel decoders. Each channel is reconstructed independently from the shared latent via $\delta_c : \mathbb{R}^{d} \to \mathbb{R}^{128} \to \mathbb{R}^{128} \to \mathbb{R}^{F}$, producing $\hat{x} \in \mathbb{R}^{B\times C\times F\times T}$ after reshaping.

DEM head (inference only). For fatigue estimation, the sequence Z (computed at a 30 s hop) is fed to a two–layer LSTM (hidden size $h{=}64$). The final context vector is mapped by a linear regressor to the 10 min DEM (Section 3.4). The encoder is kept fixed; only the LSTM regressor is trained for DEM.

~~**Optimization and schedule.** Parameters~~

2.6 Optimization and training schedule.

All network parameters were optimized using ~~Adam (Kingma and Ba, 2014) (~~the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate ~~$5{\times}10^{-3}$)~~of 5×10^{-3}. The learning rate was adapted ~~by~~ using a *ReduceLROnPlateau* scheduler (reduction factor 0.2, patience 5, minimum learning rate 10^{-5})~~based~~, monitored on the validation reconstruction loss. A batch size of 1024 was used throughout. To ensure stable optimization, gradient clipping ~~(max–norm~~ with a maximum ℓ_2 norm of 1.0 ~~), mixed precision,~~ and early stopping ~~(patience~~ with a patience of 50 ~~) were employed. The composite objective followed~~ epochs were applied.

Training followed a staged optimization strategy under the composite objective in Eq. (13)~~under a staged schedule: (i) a warm–up using only $\mathcal{L}_{\text{rec}}$ for $t_{\text{warm}}{=}100$ epochs; (ii) activation of the domain–adversarial loss $\mathcal{L}_{\text{dom}}$ with GRL scale $\gamma{=}0.4$ and a ramp $\lambda(t)$ increasing until reaching $\lambda_{\max}{=}1$; and (iii) initialization of DEC centroids by~~ . First, the autoencoder was trained using the reconstruction loss $\mathcal{L}_{\text{rec}}$ alone up to epoch $t_{\text{start,dann}}$, allowing the encoder–decoder pair to learn a stable reconstruction manifold that captures the dominant vibration patterns. The evolution of the reconstruction loss during this phase is discussed in Section 3.1.

At epoch $t_{\text{start,dann}}$, the domain-adversarial objective $\mathcal{L}_{\text{dom}}$ was activated with a gradient reversal scale $\gamma = 0.4$. Its weighting coefficient $\lambda(t)$ was then increased linearly from 0 to $\lambda_{\max}$ over the subsequent $t_{\text{duration,dann}}$ epochs. This progressive activation forces the encoder to suppress turbine-specific information while preserving the operational structure learned during the reconstruction phase.

The DEC objective was introduced at epoch $t_{\text{start,dec}}$, after initializing the cluster centroids using k~~–means followed by the introduction of $\mathcal{L}_{\text{DEC}}$ with a ramp~~ -means with $n_{\text{clusters}} = 5$. The clustering weight $\beta(t)$ ~~increasing until reaching $\beta_{\max}{=}10$. This sequencing was used to avoid competition between objectives and to prevent premature collapse of the latent manifold.~~ was increased linearly from 0 to $\beta_{\max}$ over $t_{\text{duration,dec}}$ epochs. Activation of the DEC objective reshapes the latent space to promote compact and separable operational regimes. As observed in Section 3.1, this stage may induce a transient increase in reconstruction error as the latent geometry reorganizes, followed by recovery as the decoder adapts.

Optimization. The model was trained using the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 5×10^{-3} and a *ReduceLROnPlateau* scheduler that reduced the rate by a factor of 0.2 after five epochs without improvement (minimum learning rate 10^{-5}). Batch size was set to 1024, and gradient clipping (max norm 1.0) was applied to ensure training stability . This staged optimization mitigates gradient interference between objectives that impose competing constraints on the latent space. Reconstruction training establishes a physically meaningful and stable embedding, domain-adversarial regularization removes turbine-specific bias without collapsing this structure, and clustering discretizes an already well-formed representation into interpretable operational regimes. In this order, each objective refines an existing representation rather than competing to define it from scratch, leading to improved training stability and interpretability.

All models were trained for up to 3000 epochsusing mixed-precision computation for efficiency, with early stoppingafter 50 epochs of stagnating validation loss. This adaptive optimization setup proved crucial for balancing reconstruction, adversarial, and clustering objectives. 1000 epochs, although convergence was typically achieved earlier due to early stopping. In practice, achieving stable training depended more on selecting an appropriate training stability was found to depend more strongly on batch size and learning rate schedule than on tuning the network architecture itselflearning-rate scheduling than on fine-grained architectural choices.

2.7 Evaluation methodology

For training, a random subset of 1,000 operating hours was selected per turbine hours per turbine was selected from the year 2023. 2023, corresponding to approximately six weeks of data per turbine. Each turbine was assigned an anonymized identifier(ID), and only the odd-numbered 11 out of the 44 turbines were used for model training, corresponding to half one quarter of the fleet. This partitioning split was adopted to mitigate limit overfitting and to ensure generalization across explicitly assess generalization to unseen turbines. Model testing was conducted using for operational-state inference was conducted on data from the first two weeks of 2024, which served as a constitute a temporally disjoint hold-out calibration dataset dataset used exclusively for evaluation and visualization. For the fatigue-related task, data from June to September 2024 were used to evaluate the modelfor testing, as this period includes numerous contains a high number of start–stop events and diverse a wide range of operational conditions. Since high-frequency SCADA labels are unavailable, low-frequency SCADA signals (mean power, rotor speed, pitch, wind speed), assumed constant within each 10-minute interval, are used as a reference for evaluation. Under this assumption, a lower-bound estimate of how well the embeddings capture operational information is obtained.

After training, two key aspects are examined: (i) whether the learned embeddings eliminated turbine-specific fingerprints and achieve invariance across turbines, and (ii) whether the embeddings remain informative about the underlying operational state.

2.7.1 Turbine invariance.

A key objective is to verify that the embeddings are not dominated by turbine-specific fingerprints. A straightforward option is to train a classifier to predict turbine identity from the embeddings, but the outcome of this test depends on the chosen classifier. To avoid this dependency, we adopt an information-theoretic approach and quantify the *mutual information* (MI)

between turbine identity T and the embedding $\mathbf{Z}$:

$$\mathrm{MI}(T;\mathbf{Z}) = \int\int p(t,\mathbf{z}) \log \frac{p(t,\mathbf{z})}{p(t)p(\mathbf{z})}\, dt\, d\mathbf{z}, \tag{14}$$

where $p(t,\mathbf{z})$ denotes the joint distribution of T and $\mathbf{Z}$.

Since T has 44 classes, the global MI quantifies—in bits—the total information contained in the embeddings about turbine identity, with an upper bound of $\log_2(44) \approx 5.46$ bits. This bound corresponds to a uniform distribution over turbines, which we approximate by randomly sampling 10 000 embeddings per turbine. While this single scalar captures overall dependence, it does not reveal how *individual turbines* relate to one another. To examine this structure, we compute pairwise MI. For each turbine pair (i,j), the dataset is restricted to samples from turbines i and j, the identity variable is recoded as binary $T_{ij} \in \{i,j\}$, and we estimate

$$\mathrm{MI}(T_{ij};\mathbf{Z}) \leq 1 \quad \text{bit.}$$

Pairwise MI measures how distinguishable the embeddings of two turbines are: - $\mathrm{MI}(T_{ij};\mathbf{Z}) \approx 0$ indicates nearly indistinguishable embeddings, suggesting similar dynamics; - values approaching 1 bit indicate strong separability, suggesting systematic differences. This pairwise MI inform us about the upper bound of the classification. By arranging all values into a symmetric matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$ with entries $[\mathbf{D}]_{ij} = \mathrm{MI}(T_{ij};\mathbf{Z} \mid \text{turbines } i,j)$(units: bits), a turbine similarity map is obtained. Here, $\mathrm{MI}(T_{ij};\mathbf{Z} \mid \text{turbines } i,j)$ denotes the mutual information computed using only samples from turbines i and j. This map can be interpreted in two ways:

1. **Fleet-wide dynamic clustering.** Without adversarial training (no DANN), the map highlights clusters of turbines with similar dynamics, visible as blocks of consistently low MI values within subgroups. This is useful for grouping turbines that operate under comparable dynamic conditions.

2. **Global invariance check.** With adversarial training (DANN), turbine-specific fingerprints are suppressed: matrix entries shift toward lower MI values, indicating reduced separability by turbine identity. Therefore, the same post-hoc analysis can be applied to all the turbines.

Thus, pairwise MI not only indicates how effectively DANN suppresses turbine-specific signatures, but also uncovers a data-driven similarity structure across the fleet, which is valuable for population-based SHM and cross-turbine comparisons.

2.7.2 Operational informativeness.

The second question concerns whether operational information is preserved in the embeddings. Several evaluation strategies can be considered: (i) correlations with SCADA signals, (ii) training regressors to predict SCADA from embeddings and reporting ~~R^2~~ R^2, or (iii) the use of an information-theoretic measure. For consistency, the latter approach is adopted, and the Normalized Mutual Information (NMI) between embeddings and each SCADA variable ~~S~~ S is computed:

$$\mathrm{NMI}(S;\mathbf{Z}) = \frac{\mathrm{MI}(S;\mathbf{Z})}{\sqrt{H(S)\,H(\mathbf{Z})}}, \tag{15}$$

where $H(\cdot)$ denotes Shannon entropy. Normalization ensures comparability across continuous variables by scaling MI relative to the entropies of S and $\mathbf{Z}$. As with MI, NMI is estimated using miller-madow entropy estimators as implemented in Büth et al. (2025).

In practice, 10,000 embeddings are randomly sampled per turbine from the training set to compute MI and NMI. The resulting metrics are used to jointly quantify (i) turbine invariance and (ii) operational informativeness, thereby providing a robust, model-free assessment of the learned representations. Because labeled annotations of transient events are not available, direct evaluation of event-detection performance is not feasible; instead, goodness is assessed indirectly via alignment with SCADA variables.

2.7.3 Qualitative Visualization:

UMAP dimensionality reduction (McInnes et al., 2018) was applied to project 6-dimensional embeddings into 2D space for visualization. The projections were colored according to SCADA variables and turbine identity so that both operational structure and cross-turbine consistency could be assessed.

3 Results and Discussion

The learned embeddings are evaluated along four dimensions: (i) preservation of operational information with concurrent suppression of turbine-specific signatures, (ii) generalization to unseen turbines achieved through domain-adversarial training, (iii) discretization of the ~~latent space~~ latent-space into interpretable regimes consistent with classical operational states, and (iv) predicting fatigue through the learned embedding as a replacement to the classical SCADA-based models.

~~In the following, the first two aspects are examined~~ The training dynamics and loss evolution are analyzed first in Sect. ~~3.2~~3.1, followed by the ~~analysis of regime discretization~~ assessment of turbine invariance and operational informativeness in Sect. ~~3.3 and the evaluation~~ 3.2. Regime discretization is examined in Section. 3.3, and the presence of fatigue-related information in ~~Sect~~the latent-space is evaluated in Section. 3.4.

3.1 Training dynamics and loss evolution

Figure 3 summarizes the evolution of the loss components during training under the staged optimization strategy described in Section 2.2. During the warm-up phase (epochs 0–30), only the reconstruction loss $\mathcal{L}_{\text{rec}}$ is optimized. The rapid decrease and subsequent stabilization of this loss indicate that the autoencoder learns a consistent reconstruction manifold before additional objectives are introduced.

At $t_{\text{start,dann}} = 30$, the domain-adversarial objective is activated. Its weighting coefficient λ is increased linearly over $t_{\text{duration,dann}} = 100$ epochs until it reaches $\lambda_{\max} = 0.8$. Immediately after introduction, the domain-classification loss $\mathcal{L}_{\text{domain}}$ drops sharply. This transient behavior reflects the ability of the newly trained domain classifier f_{dom} to exploit turbine-specific information still present in the latent representation. As training progresses and the gradient reversal mechanism becomes effective, the encoder increasingly suppresses turbine identity, causing the domain-classification loss to rise.

With 11 training turbines, random guessing corresponds to a cross-entropy of $\log(11) \approx 2.4$. As shown in Fig. 3, the domain-classification loss stabilizes at 2.1 The observed plateau therefore indicates that turbine identity becomes increasingly difficult to infer from the latent-space, although weak residual turbine-specific structure remains. This behavior is consistent with the objective of domain-adversarial training.

The clustering objective is introduced at $t_{\text{start,dec}} = 60$. Its weighting coefficient β is increased linearly over $t_{\text{duration,dec}} = 100$ epochs until it reaches $\beta_{\max} = 10$. Upon activation, the clustering loss $\mathcal{L}_{\text{DEC}}$ initially takes high values, reflecting the absence of well-formed clusters. As the latent-space is reshaped to promote compact and separable regimes, a transient increase in reconstruction error is observed, caused by a temporary mismatch between the decoder and the reorganized latent geometry. As optimization continues, the decoder adapts and the reconstruction loss decreases again.

The values of $\lambda_{\max}$ and $\beta_{\max}$ were selected heuristically. Moderate variations around these values did not qualitatively affect the results. The guiding principle was to scale the different loss terms at their max to be to comparable magnitudes once the reconstruction loss had stabilized. Excessively large values were found to be detrimental. In particular, setting $\beta_{\max} = 50$ led to a significant degradation of reconstruction quality, indicating excessive distortion of the latent-space.

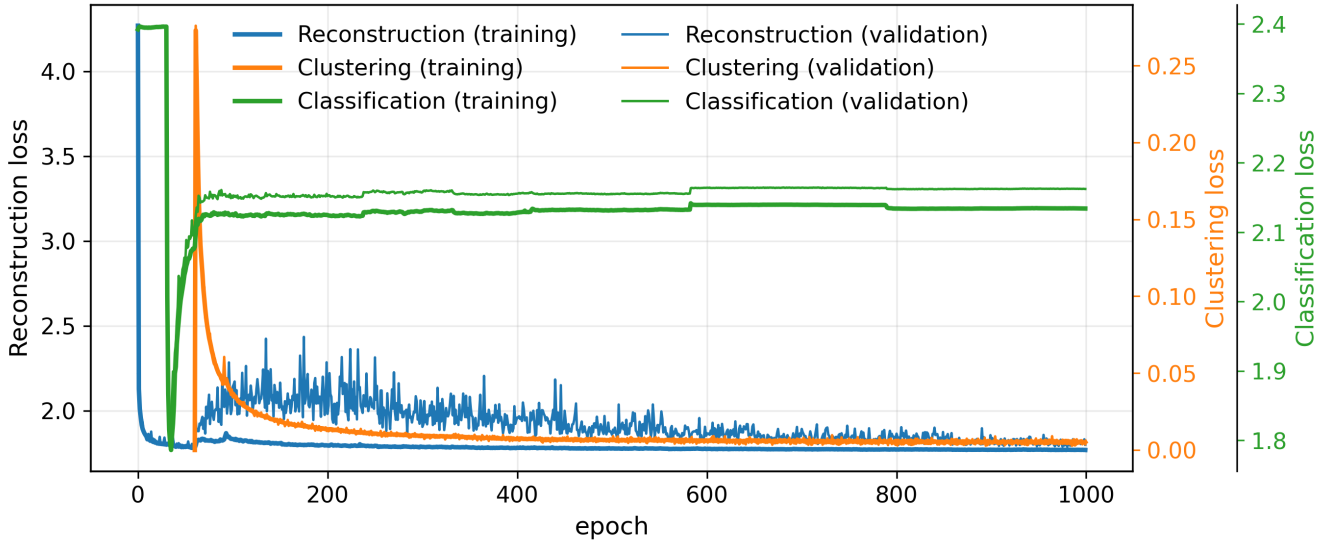

Figure 3. Training and validation evolution of the loss components: reconstruction loss $\mathcal{L}_{\text{rec}}$, domain-classification loss $\mathcal{L}_{\text{domain}}$, and clustering loss $\mathcal{L}_{\text{DEC}}$.

3.2 Assessment of Turbine Invariance and Operational Informativeness

This part focuses on the first two dimensions of evaluation. Turbine invariance is quantified by means of pairwise mutual information (MI) between turbine identity and the latent embeddings, while operational informativeness is evaluated through

normalized mutual information (NMI) between embeddings and key SCADA variables—namely power, rotor speed, pitch angle, and wind speed.

3.2.1 Turbine invariance via pairwise MI.

Turbine invariance was assessed by comparing two models: a plain autoencoder without adversarial training ($\gamma = 0$) and the same autoencoder with a domain-adversarial component applied to the ~~latent space~~ latent-space ($\gamma = 0.4$). The corresponding pairwise MI matrices, $\mathbf{D}^{(0)}$ and $\mathbf{D}^{(0.4)}$, are presented in Figs. 4 and 5.

In the absence of DANN, elevated MI values were observed for many turbine pairs (Fig. 4), indicating that turbine-specific fingerprints were retained in the embeddings alongside operational content. Subgroups of turbines were seen to be more similar to each other than to the remainder of the fleet, consistent with residual structural or site variability encoded in the ~~latent space~~latent-space.

With DANN, pairwise MI values were reduced across the matrix $\mathbf{D}^{(0.4)}$ (Fig. 5), showing that turbine identity was suppressed while operational features were preserved. Two turbines (IDs 28 and 39) remained more separable than the rest, which is interpreted as genuinely distinct dynamics rather than a training artifact. No checkerboard pattern indicative of leakage from the odd–even train/test split was observed. A small increase in reconstruction error was induced by the adversarial term, but downstream use was not compromised.

The GRL scale γ was selected by scanning $\{0.01, 0.2, 0.4, 0.6\}$ and monitoring the mean of the pairwise MI matrix $\mathbf{D}$. The mean MI was reduced from 0.65 to 0.36, then to 0.15 and 0.12, with an elbow around $\gamma = 0.4$. Larger values did not yield meaningful gains and were found to risk latent collapse, so $\gamma = 0.4$ was adopted in the final model.

Starting from the precomputed pairwise MI matrix $\mathbf{D}$ (Fig. 4), which was interpreted as a symmetric dissimilarity measure in bits (larger values corresponding to lower similarity), agglomerative hierarchical clustering was performed, resulting in the identification of five clusters (Appendix A). A detailed explanation of hierarchical clustering can be found in Contreras and Murtagh (2015).

In Fig. 6, the geographic layout of the wind farm is shown with colors indicating the clusters obtained; numbers correspond to anonymized turbine identifiers. The map was examined to verify that the clustering was not a by-product of wake geometry or simple row positioning (front versus back turbines). No systematic alignment or consistent relation with water depth was observed. The clusters nevertheless appeared structured rather than random, yet could not be explained by straightforward spatial factors. It is therefore inferred that the grouping most likely reflects a combination of site-specific conditions, control strategies, or structural variability not captured in the available metadata.

Two turbines (anonymized IDs 28 and 39) were assigned to single-member clusters and also remained the most separable after adversarial training, which suggests that their distinct behavior arises from genuine dynamic differences rather than artifacts of the clustering procedure.

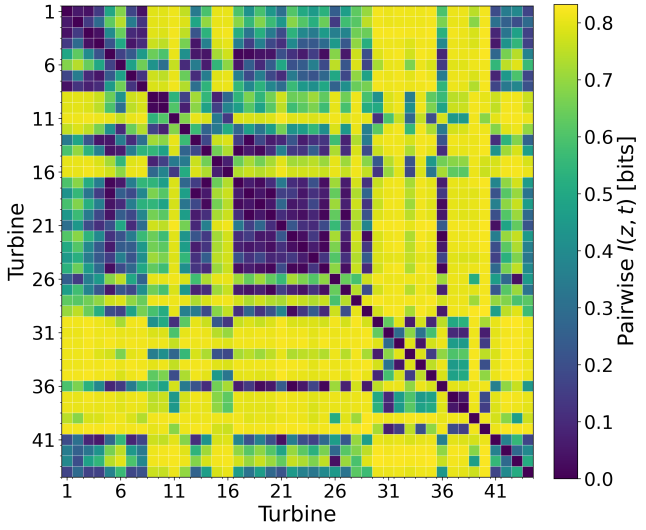

Figure 4. Pairwise MI between turbine ID and embeddings before adversarial training. Higher values indicate stronger turbine-specific signatures.

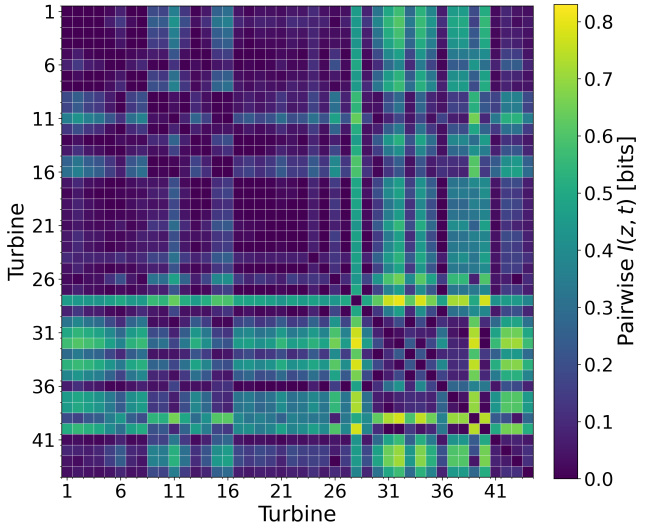

Figure 5. Pairwise MI after adversarial training (γ=0.4). Lower values indicate improved turbine invariance, while structured residuals highlight turbines with similar dynamics.

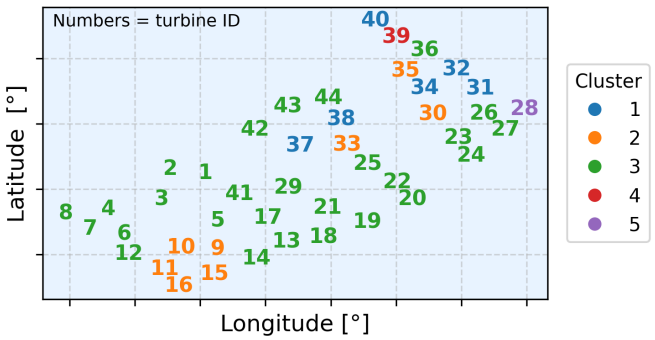

Figure 6. Wind farm layout. Numbers denote anonymized turbine IDs; color indicates clusters of the turbine based on the similarity derived from the pairwise MI in Figure4.

3.2.2 Operational informativeness via NMI.

A central objective of this study is to determine whether operational information typically derived from SCADA can instead be recovered directly from high-frequency acceleration. To evaluate this, Normalized Mutual Information (NMI) values between embeddings and SCADA variables were computed on unseen turbines and are reported in Table 1. NMI was used because it captures both linear and nonlinear dependencies and provides a normalized measure that is comparable across variables, making it well suited for assessing how much operational content is retained in the embeddings.

Across all variables, higher mean NMI values were obtained after adversarial training, with improvements ranging from $+0.016$ for power to $+0.027$ for wind speed. Values in the range of 0.75–0.92 indicate that a substantial fraction of the variability in SCADA signals can be captured by the learned embeddings, despite the fact that SCADA data were not used during training. This demonstrates that high-frequency acceleration contains operationally relevant information that can be effectively extracted through the proposed representation-learning framework.

As an external validation, random forest ~~regressor (n_{est}=100~~regressors ($n_{\text{est}} = 100$) were trained to predict SCADA variables from the learned embeddings using a 50% ~~test~~ train–test split. The models achieved mean ~~R^2~~ R^2 scores of 0.923 for power, 0.882 for pitch, 0.937 for wind speed, and 0.925 for rotor speed across the 44 turbines, ~~confirming~~ indicating that the embeddings ~~encodes operational information. Since SCADA signals were assumed constant within each~~ retain strong operationally relevant information.

Because SCADA variables are available only as 10-minute ~~interval, these values should be regarded~~ averages, any intra-interval variability captured by the vibration-based embeddings cannot be directly validated. The reported correspondence metrics therefore quantify alignment with a temporally aggregated proxy of the operational state and should be interpreted as conservative lower bounds ~~of the attainable correspondence; in other words, the model likely performs better than these metrics suggest~~with respect to the unobserved instantaneous dynamics, rather than as an upper limit on achievable predictive performance.

Table 1. Mean normalized mutual information (NMI) between embeddings and SCADA variables, computed on unseen turbines before and after adversarial training.$\Delta_{\%}$ denotes relative change.

SCADA variable	No DANN	DANN	$\Delta_{\%}$
mean_power	0.798	0.814	+2.0%
mean_pitch	0.753	0.771	+2.4%
mean_windspeed	0.919	0.946	+2.9%
mean_rpm	0.748	0.765	+2.3%

3.3 Operational state inference from embeddings

The objective in this section is to determine whether the learned ~~latent space~~ latent-space can be used to identify distinct operational states of the turbine. As shown previously, the embeddings capture SCADA-like information with high accuracy; here, the focus is on whether these representations can be organized into discrete and interpretable regimes.

When no clustering constraint is applied (~~DEC~~i.e., DEC is disabled and $\beta = 0$), the ~~latent space naturally separates into three main groups~~latent-space is shaped only by reconstruction and domain-adversarial objectives. In this setting, there is no explicit geometric incentive for the encoder to form multiple compact, well-separated operational regimes. The embedding therefore organizes primarily according to the largest, most separable dynamical differences in the data. Empirically, this yields three dominant groups, as illustrated in Fig. 9: two small clusters corresponding to non-producing conditions (standstill and parked~~conditions (differing mainly in pitch angle) and one large cluster encompassing all operating conditions. By introducing Deep Embedded Clustering (DEC))~~, and one large cluster that aggregates the full continuum of producing operation. The separation between the two non-producing clusters is consistent with a control-driven distinction (mainly pitch angle differences under low or zero rotor speed), which produces distinct low-frequency spectral signatures. In contrast, within the producing regime, sub-rated, rated, and curtailed behavior form a smooth progression in spectral space (driven by gradual changes in rotor speed, aerodynamic loading, and control action), and therefore remain embedded as a single connected manifold rather than splitting into discrete clusters. This behavior is visible in Fig. 9, where producing states form a single connected manifold the splitting is mainly driven by the pitch angle.

By introducing DEC, the latent structure is ~~refined into five clusters (a user defined number), which further distinguish between~~ explicitly encouraged to become clusterable. DEC adds a set of learnable centroids $\{\mu_j\}_{j=1}^{K}$ and optimizes the encoder such that embeddings are pulled toward these centroids via a KL-divergence objective between soft assignments and a sharpened target distribution. This explicitly trades a purely continuous representation for one that partitions the operating manifold into K compact regions. With DEC enabled and $K = 5$ (a user-defined choice), the previously broad operating manifold is refined into multiple regimes that distinguish different levels of power production ~~within the operational regime~~and control action. This five-cluster configuration aligns with the canonical division of turbine behavior ~~commonly~~ used in SCADA-based classification~~. Additionally, through DEC no need to perform a post-hoc clustering as the cluster center are defined inside~~, while being inferred directly from vibrations and at higher temporal resolution.

Additionally, DEC integrates clustering into the training objective: centroids are part of the model and ~~identified during the training~~are refined jointly with the encoder during optimization. In that sense, regime discovery is learned end-to-end rather than imposed only as a purely post-hoc clustering step on the final embeddings (although, as standard in DEC, centroids are initialized from a preliminary clustering such as k-means before being refined during training).

Figure 7 shows the resulting latent-space partition for the first five turbines during the initial two weeks of 2024, which serve as the calibration dataset (Section 2.7). The identified clusters align clearly with rotor speed thresholds, as shown in Fig. 8.

Overall, the discovered clusters correspond to well-known operating behaviours: parked/idling, standstill, sub-rated, and rated generation. Collapsing the latent geometry into discrete states provides an interpretable layer on top of the embeddings,

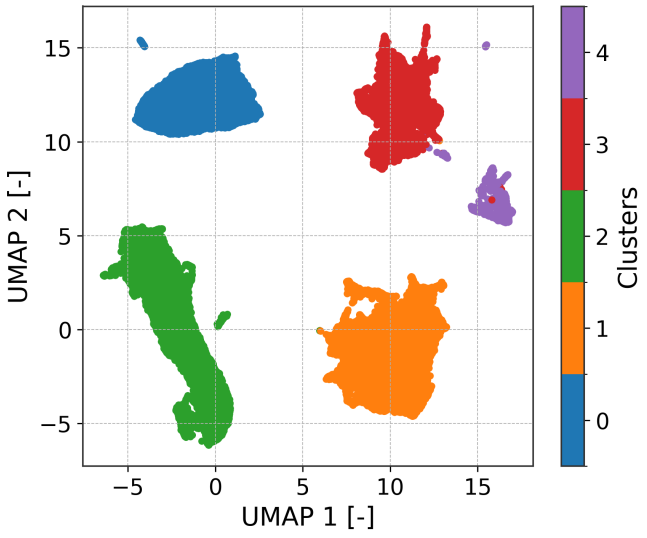

Figure 7. UMAP projection of the ~~latent space~~ latent-space coloured by discovered clusters. The partitioning yields coherent operational regimes.

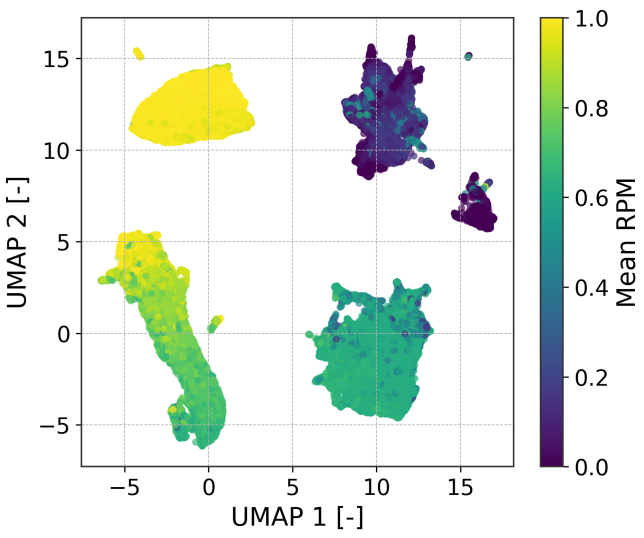

Figure 8. UMAP projection coloured by normalised mean RPM. The smooth gradient indicates that rotor speed is preserved in the embedding.

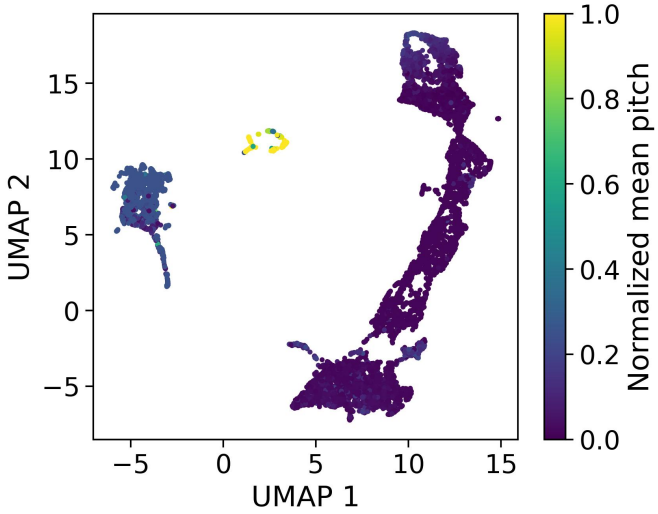

Figure 9. UMAP projection of the latent embeddings learned without Deep Embedded Clustering (DEC), coloured by normalised mean blade pitch. The representation separates parked and standstill conditions from operating states, but the operating regime remains a largely continuous manifold without clear sub-structure. This illustrates that, in the absence of an explicit clustering objective, the latent-space is not naturally partitioned into distinct operational regimes, motivating the use of DEC to enforce clusterable and interpretable embeddings.

enabling event monitoring tasks such as start/stop counting at sub–10-minute resolution. In scenarios without SCADA, clusters can be interpreted by visually inspecting representative samples; once identified, they can be relabelled with meaningful operational states.

To further validate the clustering, the regimes are projected onto SCADA references. In the power curve (Fig. 10), the regimes separate into five operating zones. Clusters 3 and 4 overlap at low power, but their distinction becomes clear on the pitch versus wind-speed plot (Fig. 11), where cluster 4 corresponds to high pitch (curtailed or stopped) and cluster 3 to lower pitch (idling). A small overlap is also observed between the rated and ramp-up regions, reflecting ~~there similarity in term~~ their similarity in terms of RPM and ~~there spectrais similar~~the resulting spectra. These comparisons should be regarded as a lower-bound validation, since SCADA signals are available only as 10-minute averages, whereas embeddings are computed at a 30-second hop length. The assumption of constant SCADA over 10 minutes introduces unavoidable mismatches.

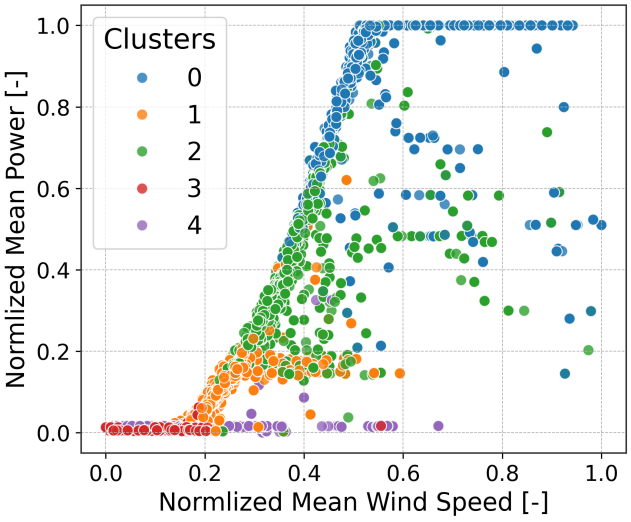

Figure 10. Normalised power versus wind speed, coloured by latent-space clusters. Regimes align with canonical power-curve regions.

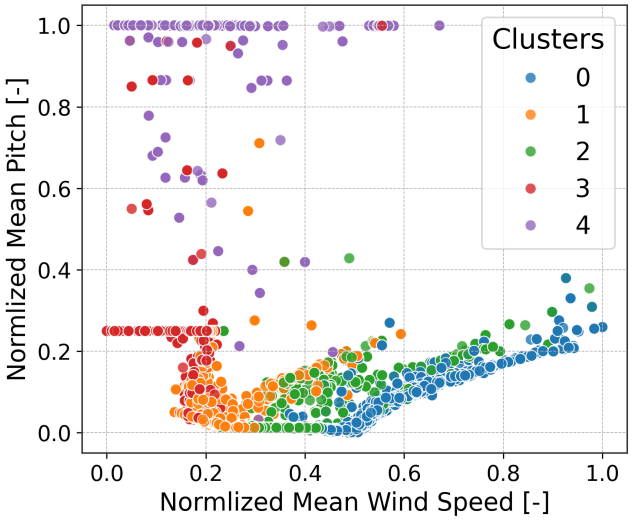

Figure 11. Normalised pitch versus wind speed, coloured by clusters. High-pitch curtailed/stopped regimes separate from low-pitch operating regimes.

Finally, the method enables monitoring of high-frequency operational events. By applying the model continuously, it is possible to identify and count start/stop transitions within each hour, capturing short events that are lost in coarse 10-minute SCADA averages. Figure 12 illustrates such a case: six stop–start events are detected, with transitions from low-rate production to standstill. These rapid fluctuations have direct implications for fatigue life, underlining the value of high-resolution, acceleration-based regime inference.

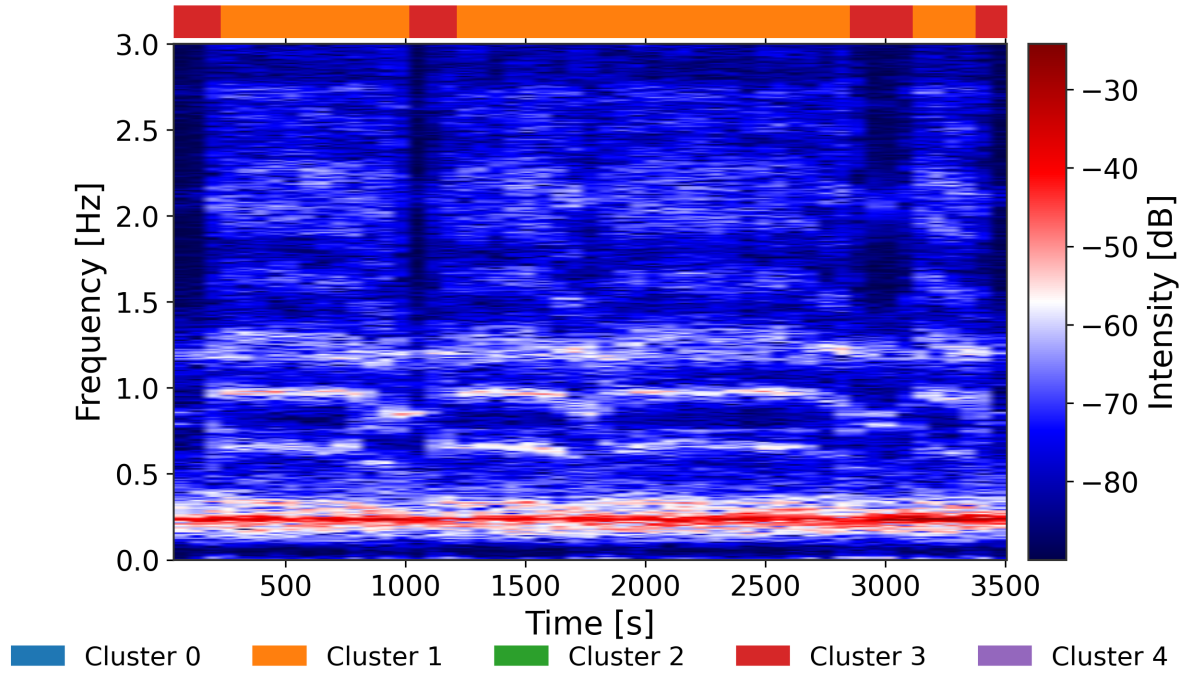

Figure 12. Example of high-frequency event detection from embeddings. ~~Four~~ six stop–start transitions are resolved within one hour, highlighting dynamic loading conditions that would be obscured in 10-minute SCADA.

3.4 Damage estimation from embeddings

~~Operational insight~~ Operational-state information is fundamental for ~~many aspects of wind turbine management, particularly for assessing fatigue damage. Traditionally,~~ wind turbine fatigue assessment. Damage Equivalent Moment (DEM) estimation ~~has relied on~~ is traditionally performed using SCADA-based models ~~, which are constrained by their coarse~~ trained on 10-minute ~~sampling and by the limited availability~~ averages and calibrated on a limited number of strain-instrumented reference turbines. In this ~~section, the method of (de N Santos et al., 2024) is employed to estimate DEM but replacing the use of SCADA with directly using acceleration-derived embeddings to~~ work, DEM estimation is used as an auxiliary validation task to assess whether the learned ~~representations~~ acceleration-derived embeddings preserve load-relevant information.

~~Since the acceleration~~

3.4.1 Baseline definition

In this study, the proposed method is compared against a predefined reference baseline. This baseline corresponds to the artificial neural network–based fatigue estimation framework introduced in de N Santos et al. (2024). It combines standard 10-minute SCADA variables with acceleration-based features derived from simple statistical descriptors of vibration signals

(e.g. RMS, variance, and standard deviation). The baseline reflects current state-of-practice for farm-wide DEM estimation and is actively used in operational deployments

Importantly, the comparison is asymmetric by design. The baseline model is trained on a substantially longer dataset, namely a full year of data (and two full years in the internal implementation). In contrast, the proposed approach is trained on only 1,000 hours per turbine and does not use SCADA information at any stage, although it takes full leverage of the higher sampling rate. The baseline, as such, constitutes a best-case reference rather than an information-parity comparator.

Therefore, the current work does not attempt to compare a SCADA-only against an acceleration-only approach, but rather position our approach in relation to the current state-of-the-art (which uses SCADA and acceleration statistics). In a previous study (albeit undertaken for a different wind farm with different foundations), we have demonstrated how the addition of 10-minute acceleration statistics to SCADA-only models improves fatigue prediction accuracy by several percentage points (d N Santos et al., 2022). It is therefore against this baseline that we benchmark our approach.

3.4.2 Training and evaluation protocol

Acceleration data are stored in ~~1-hour fileswhile~~ one-hour files, whereas DEM values are available ~~in~~ at 10-minute ~~intervals, each~~ resolution. Each acceleration file is ~~divided~~ segmented into six non-overlapping 10-minute ~~segments. Each segment is then~~ intervals, each associated with the corresponding DEM value computed over the same time window. ~~The encoder was~~

The encoder is kept fixed, and only ~~the~~ an LSTM-based regression head ~~was trained , following~~ is trained to map sequences of embeddings to DEM values. Training and evaluation follow a train-on-4 / test-on-1 cross-validation strategy across the five strain-instrumented ~~fleet leaders. To contextualize performance, a legacy SCADA-based baseline was compared against the proposed acceleration-only approach. However, this comparison remains asymmetrical, as the baseline model was trained on substantially more data and used all five Fleet Leaders (FLs)for training~~fleet-leader turbines; in the reported setup, the model is trained on FL1–FL4 and evaluated on FL5, which is entirely unseen during training. Evaluation is performed exclusively over the full summer period (June–September 2024), which includes a wide range of operational conditions and numerous stop–start events.

~~Across all~~

3.4.3 Results on the downstream fatigue estimation task

Across all fleet-leader turbines, the proposed ~~acceleration-based approach achieved predictive power exceeding that of the baseline model. The baseline combines SCADA variables with handcrafted features derived from acceleration data and is referred to here simply as the SCADA~~ acceleration-only approach achieves predictive performance that is comparable to or exceeds that of the SCADA-based baseline. Differences in R^2 typically remain within a narrow range of 0.01~~–0.02, which is meaningful given that values near 1 represent the upper performance limit, thus getting closer to 1 is more and more challenging. In terms of~~ –0.02, while mean squared error ~~(MSE), the proposed model performs better overall, despite relying on fewer sensors and substantially less training data . These~~ is consistently similar or lower.

Given the strong advantage of the baseline in terms of training data volume and sensor availability, these results indicate that the learned embeddings preserve sufficient load-related information to ~~enable fleet-wide~~ support fatigue estimation without ~~dependence on SCADA~~ reliance on SCADA variables or manually engineered features. ~~The baseline approach corresponds to the method described in de N Santos et al. (2024). The presented results are based on 4 months of data collected during the summer of 2024. The summer often contains more variability in term of operational condition~~

~~Moreover, since we enforced the embedding to be similar for all turbine and through the pairwise mutual information table we showed Figure 5we can feel more comfortable deploying the model to the whole windfarm and estimating fatigue except for turbine ID~~ Finally, fleet-wide applicability is supported by the turbine-invariant representations learned through domain-adversarial training. Pairwise mutual information analysis between embeddings and turbine identity (Fig. 5) shows that turbine-specific information is largely suppressed for most units. Turbines 28 and ~~39.~~ 39 exhibit residual turbine-specific behavior and are therefore excluded from fleet-level fatigue estimation.

Table 2. DEM prediction on all strain-instrumented turbines: comparison between the SCADA-based legacy baseline and the proposed acceleration-only embedding approach. Each turbine was unseen during training (train-on-4 / test-on-1). R^2 higher is better, MSE lower is better

Turbine	**Model**	$\mathbf{R^2}$ $(\uparrow)$	**MSE** $(10^{10}, \downarrow)$
FL1	SCADA baseline	0.95	2.1
	Proposed approach	0.97	1.5
FL2	SCADA baseline	0.95	1.6
	Proposed approach	0.97	1.4
FL3	SCADA baseline	0.94	2.9
	Proposed approach	0.95	2.8
FL4	SCADA baseline	0.96	2.5
	Proposed approach	0.96	2.4
FL5	SCADA baseline	0.95	1.7
	Proposed approach	0.96	1.5

4 Conclusions

This study has demonstrated that high-frequency nacelle acceleration can serve as a reliable foundation for inferring wind turbine operational state when SCADA is unavailable, incomplete, or too coarse. By learning compact, turbine-invariant embeddings of short-time spectrograms, the proposed framework captured operational dynamics at sub–10-minute resolution and aligned closely with supervisory variables, despite never being trained on SCADA. Domain-adversarial training effectively reduced turbine-specific bias, enabling consistent cross-turbine structure and supporting deployment across a mainly homogeneous fleet without per-turbine training.

Discrete operational regimes derived from the embeddings provided an interpretable bridge to classical power-curve analysis, allowing events such as starts, stops, and curtailments to be resolved at finer temporal scales than is possible with standard SCADA. In an auxiliary illustration, sequences of embeddings were further shown to predict damage-equivalent moments (DEM) with competitive accuracy relative to a SCADA-based baseline, demonstrating that acceleration-derived representations can preserve load-relevant information needed for fatigue-related applications.

While fatigue estimation was not the primary focus of this work, these results indicate that the operational embeddings retain physically meaningful variability beyond regime identification. A dedicated investigation of the interaction between domain-adversarial regularization and fatigue prediction—quantifying the trade-off between turbine invariance and preservation of load- and site-specific effects such as soil–structure interaction—remains an important direction for future research. Such a study would require a broader set of strain-instrumented turbines and is therefore left for future work.

Together, these findings establish acceleration-based operational embeddings as a practical and scalable complement to SCADA for structural health monitoring and performance analysis. While the present validation was performed on a single offshore farm, the results suggest a broader potential: cross-farm transfer, integration of physics-informed constraints, and tighter coupling of embeddings to load proxies are promising directions for future research. By leveraging ubiquitous accelerometers and modern representation learning, SCADA-free monitoring becomes a viable path toward richer, higher-resolution insight into turbine dynamics, unlocking new opportunities for condition assessment, fatigue extrapolation, and predictive maintenance across large wind fleets.

Author contributions. Y. Bel-Hadj led the conceptualization, methodology design, software implementation, data curation, formal analysis, and original draft preparation. F. de Nolasco Santos contributed to conceptualization, methodology design, and critical revision of the manuscript. W. Weijtjens contributed to conceptualization, validation, and supervision. C. Devriendt provided resources, supervision, and project administration. All authors contributed to manuscript review and editing.

Competing interests. No conflict of interest is declared.

Financial support The authors gratefully acknowledge the support from Flanders Innovation & Entrepreneurship (VLAIO) through the Supersized 5.0 research project. The present research work is also part of the WILLOW project, Funded by the European Union with GA No. 1011122184. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them

Declaration of Generative AI and AI-assisted technologies The authors used ChatGPT (version GPT-5) during the preparation of this work to improve the language. These tools were used to streamline the writing process, but not to generate or interpret scientific content. All AI-assisted content was reviewed, edited, and verified by the authors as needed.

References

IEC 61400-25-6: Communications for monitoring and control of wind power plants – Logical node classes and data classes for condition monitoring, 2016.

IEC 61400-1: Wind energy generation systems – Part 1: Design requirements, 2019.

Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., and Marchand, M.: Domain-adversarial neural networks, arXiv preprint arXiv:1412.4446, https://arxiv.org/abs/1412.4446, 2014.

Avendano-Valencia, L. D., Chatzi, E. N., and Tcherniak, D.: Gaussian process models for mitigation of operational variability in the structural health monitoring of wind turbines, Mechanical Systems and Signal Processing, 142, 106 686, 2020.

Bel-Hadj, Y. and Weijtjens, W.: Anomaly detection in vibration signals for structural health monitoring of an offshore wind turbine, in: European Workshop on Structural Health Monitoring, pp. 348–358, Springer, 2022.

Bel-Hadj, Y., Weijtjens, W., and de Nolasco Santos, F.: Anomaly detection and representation learning in an instrumented railway bridge., in: ESANN, 2022.

Bel-Hadj, Y., Weijtjens, W., and Devriendt, C.: Structural health monitoring in a population of similar structures with self-supervised learning: a two-stage approach for enhanced damage detection and model tuning, Structural Health Monitoring, p. 14759217251324194, 2025.

Bengio, Y., Courville, A., and Vincent, P.: Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence, 35, 1798–1828, 2013.

Bette, H. M., Wiedemann, C., Wächter, M., Freund, J., Peinke, J., and Guhr, T.: Dynamics of wind turbine operational states, arXiv preprint arXiv:2310.06098, 2023.

Bull, L. A., Gardner, P. A., Gosliga, J., Dervilis, N., Papatheou, E., Maguire, A. E., Campos, C., Rogers, T. J., Cross, E. J., and Worden, K.: Towards population-based structural health monitoring, Part I: Homogeneous populations and forms, in: Model Validation and Uncertainty Quantification, Volume 3: Proceedings of the 38th IMAC, A Conference and Exposition on Structural Dynamics 2020, pp. 287–302, Springer, 2020.

Büth, C. M., Acharya, K., and Zanin, M.: infomeasure: a comprehensive Python package for information theory measures and estimators, Scientific Reports, 15, 29 323, 2025.

Byrne, B. W., Burd, H. J., Zdravković, L., McAdam, R. A., Taborda, D. M., Houlsby, G. T., Jardine, R. J., Martin, C. M., Potts, D. M., and Gavin, K. G.: PISA: new design methods for offshore wind turbine monopiles, Revue Française de Géotechnique, p. 3, 2019.

Chu, J.-c., Yuan, L., Xie, F., Pan, L., Wang, X.-d., and Zhang, L.-z.: Operational State Analysis of Wind Turbines Based on SCADA Data, in: 2nd International Conference on Electrical and Electronic Engineering (EEE 2019), pp. 169–173, Atlantis Press, 2019.

Contreras, P. and Murtagh, F.: Hierarchical clustering, Handbook of cluster analysis, pp. 103–123, 2015.

Cooley, J. W. and Tukey, J. W.: An algorithm for the machine calculation of complex Fourier series, Mathematics of computation, 19, 297–301, 1965.

d N Santos, F., Noppe, N., Weijtjens, W., and Devriendt, C.: Data-driven farm-wide fatigue estimation on jacket-foundation OWTs for multiple SHM setups, Wind Energy Science, 7, 299–321, https://doi.org/10.5194/wes-7-299-2022, 2022.

Daems, P.-J., Peeters, C., Matthys, J., Verstraeten, T., and Helsen, J.: Fleet-wide analytics on field data targeting condition and lifetime aspects of wind turbine drivetrains, Forschung im Ingenieurwesen, 87, 285–295, 2023.

de N Santos, F., Noppe, N., Weijtjens, W., and Devriendt, C.: Data-driven farm-wide fatigue estimation on jacket foundation OWTs for multiple SHM setups, Wind Energy Science Discussions, 2021, 1–36, 2021.

de N Santos, F., D'Antuono, P., Robbelein, K., Noppe, N., Weijtjens, W., and Devriendt, C.: Long-term fatigue estimation on offshore wind turbines interface loads through loss function physics-guided learning of neural networks, Renewable Energy, 205, 461–474, 2023.

de N Santos, F., Noppe, N., Weijtjens, W., and Devriendt, C.: Farm-wide interface fatigue loads estimation: A data-driven approach based on accelerometers, Wind Energy, 27, 321–340, 2024.

de Nolasco Santos, F., Bel-Hadj, Y., Weijtjens, W., and Devriendt, C.: Estimating Fatigue Through Latent Space Embedding of Acceleration in Offshore Wind Turbines, in: International Conference on Experimental Vibration Analysis for Civil Engineering Structures, pp. 943–951, Springer, 2025.

Ganin, Y. and Lempitsky, V.: Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, pp. 1180–1189, PMLR, 2015.

Gardner, P., Bull, L. A., Gosliga, J., Poole, J., Dervilis, N., and Worden, K.: A population-based SHM methodology for heterogeneous structures: Transferring damage localisation knowledge between different aircraft wings, Mechanical Systems and Signal Processing, 172, 108 918, 2022.

Ha, D. and Schmidhuber, J.: Recurrent world models facilitate policy evolution, Advances in neural information processing systems, 31, 2018.

Hameed, Z., Hong, Y. S., Cho, Y. M., Ahn, S. H., and Song, C. K.: Condition monitoring and fault detection of wind turbines and related algorithms: A review, Renewable and Sustainable Energy Reviews, 13, 1–39, 2009.

Hinton, G. E. and Salakhutdinov, R. R.: Reducing the dimensionality of data with neural networks, Science, 313, 504–507, https://doi.org/10.1126/science.1127647, 2006.

Hlaing, N., Morato, P. G., Santos, F. d. N., Weijtjens, W., Devriendt, C., and Rigo, P.: Farm-wide virtual load monitoring for offshore wind structures via Bayesian neural networks, Structural Health Monitoring, 23, 1641–1663, 2024.

Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

Korkos, P., Linjama, M., Kleemola, J., and Lehtovaara, A.: Data annotation and feature extraction in fault detection in a wind turbine hydraulic pitch system, Renewable Energy, 185, 692–703, 2022.

LeCun, Y., Bengio, Y., and Hinton, G.: Deep learning, Nature, 521, 436–444, https://doi.org/10.1038/nature14539, 2015.

Li, Z., Liu, Y., and Xia, Y.: Damage detection of bridges subjected to moving load based on domain-adversarial neural network considering measurement and model error, Engineering Structures, 293, 116 601, https://doi.org/10.1016/j.engstruct.2023.116601, 2023.

Li, Z., Chen, Y., Xu, T., and Huang, H.: Cross-domain damage detection through partial conditional adversarial domain adaptation, Mechanical Systems and Signal Processing, 225, 110 118, https://doi.org/10.1016/j.ymssp.2025.110118, 2025.

Liu, D., Wang, T., Liu, S., Wang, R., Yao, S., and Abdelzaher, T.: Contrastive self-supervised representation learning for sensing signals from the time-frequency perspective, in: IEEE International Conference on Computer Communications (INFOCOM Workshops), pp. 1–6, IEEE, https://doi.org/10.1109/ICCCN52240.2021.9522151, 2021.

Mao, W., He, J., Li, Y., and Yan, Y.: A new structured domain adversarial neural network for transfer fault diagnosis of rolling bearings under different working conditions, IEEE Transactions on Instrumentation and Measurement, 70, 1–13, https://doi.org/10.1109/TIM.2020.3040593, 2020.

Martakis, P., Chatzi, E., Michalis, I., and Karapetrou, S.: Fusing damage-sensitive features and domain adaptation towards robust damage classification in real buildings, Soil Dynamics and Earthquake Engineering, 166, 107 739, https://doi.org/10.1016/j.soildyn.2023.107739, 2023.

McInnes, L., Healy, J., and Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, arXiv preprint arXiv:1802.03426, 2018.

Ozturkoglu, O., Ozcelik, O., and Günel, S.: Effects of Operational and Environmental Conditions on Estimated Dynamic Characteristics of a Large In-service Wind Turbine, Journal of Vibration Engineering & Technologies, 12, 803–824, 2024.

Rahimi Taghanaki, F. et al.: Self-supervised human activity recognition with localized time-frequency contrastive representation learning, in: Proceedings of the 30th ACM International Conference on Multimedia, ACM, https://doi.org/10.1145/3581783.3612063, 2023.

Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J., Le, Q. V., and Ng, A. Y.: Building high-level features using large scale unsupervised learning, in: Proceedings of the 29th International Conference on Machine Learning (ICML-12), pp. 81–88, 2012.

Singh, D., Dwight, R., and Viré, A.: Probabilistic surrogate modeling of damage equivalent loads on onshore and offshore wind turbines using mixture density networks, Wind Energy Science Discussions, 2024, 1–28, 2024.

Snover, D.: Urban Seismic Noise Identified with Deep Embedded Clustering Using a Dense Array in Long Beach, CA, Master's thesis, University of California San Diego, available from https://noiselab.ucsd.edu/group/Thesis/DSnover_MastersThesis.pdf, 2020.

Soares-Ramos, E. P., de Oliveira-Assis, L., Sarrias-Mena, R., and Fernández-Ramírez, L. M.: Current status and future trends of offshore wind power in Europe, Energy, 202, 117 787, 2020.

Tschannen, M., Bachem, O., and Lucic, M.: Recent advances in autoencoder-based representation learning, arXiv preprint arXiv:1812.05069, https://arxiv.org/abs/1812.05069, 2018.

Vincent, P.: A connection between score matching and denoising autoencoders, Neural computation, 23, 1661–1674, 2011.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion., Journal of machine learning research, 11, 2010.

Weijtens, W., Noppe, N., Verbelen, T., Iliopoulos, A., and Devriendt, C.: Offshore wind turbine foundation monitoring, extrapolating fatigue measurements from fleet leaders to the entire wind farm, in: Journal of Physics: Conference Series, vol. 753, p. 092018, IOP Publishing, 2016.

Xie, J., Girshick, R., and Farhadi, A.: Unsupervised deep embedding for clustering analysis, in: International conference on machine learning, pp. 478–487, PMLR, 2016.

Zhao, Y., Pan, J., Huang, Z., Miao, Y., Jiang, J., and Wang, Z.: Analysis of vibration monitoring data of an onshore wind turbine under different operational conditions, Engineering Structures, 205, 110 071, 2020.

Appendix A: Hierarchical clustering from the precomputed MI dissimilarity D

Clustering was performed directly on the precomputed turbine×turbine matrix **D**, which encodes pairwise dissimilarity derived from mutual information (MI) between turbine identity and embeddings. Larger entries in **D** indicate lower similarity (stronger turbine-specific signatures), the matrix is symmetric with a zero diagonal, and units are bits.

Agglomerative hierarchical clustering with *average linkage* (UPGMA) was applied to **D**. The number of clusters was determined by the *largest merge-jump* rule: the tree was cut at the midway between the two consecutive merges exhibiting the largest increase in linkage distance, yielding 5 clusters. Leaf labels were anonymized using the same mapping as in the main text. The linkage distance on the vertical axis shares the units of **D** (bits). The resulting partition is the one used to color the

geographic layout map in Fig. 6. The dendrogram below corresponds to embeddings trained without adversarial regularization ($\gamma = 0$).

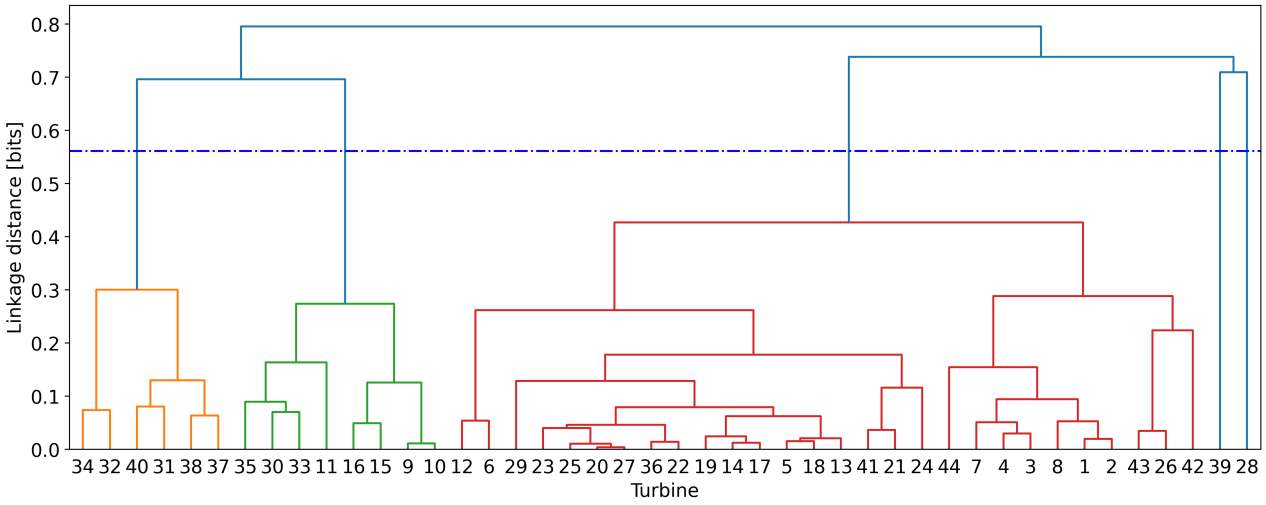

Figure A1. Dendrogram from pairwise mutual-information dissimilarity $\mathbf{D}$ between turbines based on acceleration-derived embeddings (no DANN). Each merge height reflects the dissimilarity in bits; higher values indicate more distinct turbine dynamics. The five clusters obtained correspond to groups of turbines with similar vibration behavior as represented by the autoencoder.