

## Reviewer 1 Comments

The manuscript proposes a SCADA-free methodology to infer wind turbine operational regimes and damage equivalent moments from high-frequency acceleration data. The approach relies on self-supervised representation learning using an autoencoder, combined with domain-adversarial regularization to suppress turbine-specific signatures and a clustering objective to separate operating regimes. A downstream model predicts DEM from the learned latent representation. Validation is performed using 10-minute SCADA data.

The proposed methodology is original, performs well, and addresses a relevant practical limitation. The step-by-step structure is clear and the manuscript is easy to follow. Publication is recommended after addressing the following comments.

### Comments and questions

- 1) The first paragraph of the abstract "Wind turbine operation (...) contain gaps" is repeated twice.
- 2) Line 15 of the abstract : missing a dot after "DEC facilitates the post-hoc analysis of the learned embedding".
- 3) Section 2.2.2 : Denoting the dimension of the latent space by  $L$  (line 232) is ambiguous, since  $L$  is already used for the length of the time window in 2.2.1, line 213. Inconsistency with line 240, where the dimension of latent space is denoted as  $d$ .
- 4) Line 232 :  $L <$
- 5) Section 2.2.7 : How did you choose the number of epoch  $t_{\text{warm}}$  for pretraining ? Has the loss converged after 100 epochs, or is it still decreasing ?
- 6) How does each term compare on the loss ? Do they weight equally ?
- 7) Line 320 (and line 368) says "This sequencing avoids competition between objectives". Could you clarify why?
- 8) Did you visualize how each loss changes as training progresses ? Optimizing the composite loss does not necessarily mean that all three losses are decreasing. One loss may drive the composite loss, while another one is not moving, or even increasing.
- 9) Section 2.4 : When you mention "A linear regression head", does it refer to the last layer of the LSTM network, or to an independent model ? In the latter, how is the LSTM trained ?
- 10) Line 352 : Unless I am mistaken, the abbreviation MLP (Multilayer perceptron?) is not introduced explicitly, which could confuse the reader.
- 11) Section 2.6 : Isn't two weeks a short period for testing, compared to one year for training ? Is the autoencoder performance similar in all periods of the year ?
- 12) line 488-490 : "Since SCADA signals were assumed constant within each 10-minute interval, these values should be regarded as conservative lower bounds of the attainable correspondence; in other words, the model likely performs better than these metrics suggest." Is this true ? One could argue that high frequency data is more complex and therefore more difficult to estimate from acceleration data.

- 13) Line 500 : If we were to perform post-hoc clustering, would we get clusters similar to DEC ?  
One also has to make sure that the clustering loss does not affect the latent space mapping negatively, in particular for DEM estimation which is its other application.
- 14) Line 512 : spelling mistake : "there" instead of "their" (twice).
- 15) What model is used for the SCADA baseline ?
- 16) Line 535 : "The baseline combines SCADA variables with handcrafted features derived from acceleration data". Why not using only SCADA as a baseline ? If acceleration is included, then information received by the autoencoder+LSTM approach is partially included in the baseline. As a result, the comparison evaluates the performance of the model architecture and data processing more than the difference between SCADA and acceleration data.

## Response to comment and question

### Response to Reviewer 1

We thank Reviewer 1 for the careful reading of the manuscript and for the constructive comments. The suggestions helped improve clarity, notation consistency, and transparency of the training procedure. Each comment is addressed below. All corrections and clarifications have been incorporated into the revised manuscript.

---

#### Comment 1

**The first paragraph of the abstract (“Wind turbine operation (...) contain gaps”) is repeated twice.**

#### Response

We acknowledge this duplication.

#### Change made in manuscript

The repeated paragraph was removed. The abstract now contains a single introductory paragraph, appearing only once.

---

#### Comment 2

**Line 15 of the abstract: missing a dot after “DEC facilitates the post-hoc analysis of the learned embedding”.**

#### Response

thanks for this comment .

### **Change made in manuscript**

A period was added at the end of the sentence.

---

### **Comment 3**

**Section 2.2.2: Denoting the latent-space dimension by  $L$  (line 232) is ambiguous, since  $L$  is already used for the time-window length in Section 2.2.1 (line 213). Inconsistency with line 240, where the latent dimension is denoted as  $d$ .**

### **Response**

thanks for this comment, we fixed the issue

### **Change made in manuscript**

Notation was made consistent throughout: the latent-space dimension is now denoted exclusively by  $d$ , while  $L$  is reserved for the time-window length.

---

### **Comment 4**

**Line 232: stray character “<”.**

### **Response**

Correct.

### **Change made in manuscript**

The stray “<” character was removed.

---

### **Comment 5**

**Section 2.2.7: How did you choose the number of epochs  $t_{\text{warm}}$  for pretraining? Has the loss converged after 100 epochs, or is it still decreasing?**

### **Response**

We agree that the submitted version did not document this clearly enough. To improve transparency, we expanded the description of the training schedule and added an explicit analysis of training dynamics. The revised manuscript shows that the reconstruction loss stabilizes before the domain-adversarial and clustering objectives are introduced, supporting the staged training strategy. The choice of  $t_{\text{warm\_dann}}$  and  $t_{\text{warm\_dec}}$  is heuristic, the rule was  $t_{\text{warm}}$  should be long enough to let the

reconstruction loss start stabilizing. We show how the different losses evolve through training in a new section

### **Change made in manuscript**

- Added a new **Section 3.1 (Training dynamics and loss evolution)**, including a figure showing the evolution of all loss components throughout training.
  - Reworked **Section 2.2.7** to clarify the role of  $t_{\text{warm}}$  into subsection 2.6
  - The revised text explicitly states that the reconstruction loss stabilizes before activating domain-adversarial and clustering objectives.
- 

### **Comment 6**

#### **How does each term compare in the loss? Do they weight equally?**

#### **Response**

The loss terms are not weighted equally. Their relative weights are heuristic and selected based on training dynamics rather than a principled optimum. The guiding principle is to scale the additional loss terms such that their magnitudes become comparable to the stabilized reconstruction loss, and to introduce them progressively during training.

A brief sensitivity analysis showed limited impact on final results within reasonable ranges of  $\lambda$  and  $\beta$ . However, excessively large values are detrimental because they distort the latent space.

#### **Change made in manuscript**

We explicitly specified the training schedule and the weights. The following text (now included in the revised manuscript) documents the optimization and staged schedule:

#### **Optimization and training schedule.**

All parameters were optimized using the Adam optimizer with an initial learning rate of  $5 \times 10^{-3}$ . The learning rate was adapted using a ReduceLROnPlateau scheduler (reduction factor 0.2, patience 5, minimum learning rate  $10^{-5}$ ) monitored on the validation reconstruction loss. A batch size of 1024, gradient clipping with a maximum norm of 1.0, and early stopping with a patience of 50 epochs were employed to ensure stable optimization.

Training followed a staged schedule under the composite objective in Eq (total loss) First, the autoencoder was trained using only the reconstruction loss  $\mathcal{L}_{\text{rec}}$  for  $t_{\text{warm}} = 30$  epochs to establish a stable reconstruction manifold. Second, the domain-

adversarial loss  $\mathcal{L}_{\text{dom}}$  was activated with a gradient reversal scale  $\gamma = 0.4$ , and its weighting coefficient  $\lambda(t)$  was linearly increased from 0 to  $\lambda_{\text{max}} = 0.8$  over the subsequent 100 epochs. Third, Deep Embedded Clustering (DEC) was introduced at epoch 60 after initializing the cluster centroids using  $k$ -means with  $n_{\text{clusters}} = 5$ . The clustering weight  $\beta(t)$  was then linearly ramped from 0 to  $\beta_{\text{max}} = 10$  over 100 epochs.

---

## Comment 7

**Line 320 (and line 368) says: “This sequencing avoids competition between objectives”. Could you clarify why?**

### Response

We agree that the original phrasing was imprecise. The objectives do compete once jointly optimized; staged training does not eliminate competition. Rather, it reduces strong interference during early training by ensuring that each objective is applied on a sufficiently structured latent space.

The adopted sequencing follows established practice in both Deep Embedded Clustering (DEC) and Domain-Adversarial Neural Networks (DANN): DEC commonly pretrains the autoencoder before adding the clustering constraint, and DANN commonly activates adversarial regularization after the classifier (original paper of DANN used a classifier and not an autoencoder) has learned a meaningful representation. In our case, staged activation improves stability and makes the effect of each objective easier to interpret.

### Change made in manuscript

Section 2.6 was reworked. The revised manuscript now uses the following formulation:

This staged optimization mitigates gradient interference by introducing constraints on the latent space in a progressive manner. The reconstruction objective is optimized first to establish a physically meaningful and stable embedding that captures the dominant vibration patterns. Domain-adversarial regularization is then activated to suppress turbine-specific signatures while preserving this structure. Finally, the clustering objective is introduced to discretize an already well-formed latent space into compact operational regimes. This ordering ensures that each objective refines an existing representation rather than competing to define it from scratch, which improves training stability and interpretability.

All models were trained for up to 1000 epochs, although convergence was typically achieved earlier due to early stopping. In practice, training stability was found to depend more strongly on batch size and learning-rate scheduling than on fine-grained architectural choices.

---

## Comment 8

**Did you visualize how each loss changes as training progresses? Optimizing the composite loss does not necessarily mean all three losses are decreasing.**

### Response

Indeed this is a missing part that is added now. We now explicitly visualize and discuss the evolution of all loss components throughout training. As expected, the losses do not all decrease simultaneously. In particular, the domain-classification loss increases toward the random-guess baseline as turbine-specific information is suppressed, while reconstruction remains stable after transient effects when new objectives are introduced.

### Change made in manuscript

A new Section 3.1 was added, and the following text is included verbatim in the revised manuscript:

Training dynamics and loss evolution:

Figure 3 summarizes the evolution of the loss components during training under the staged optimization strategy described in Section 2.6. 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  $\lambda$  is increased linearly over  $t_{\text{duration,dann}} = 100$  epochs until it reaches  $\lambda_{\text{max}} = 0.8$ . 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  $f_{\text{dom}}$  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 1 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  $\beta$  is increased linearly over  $t_{\text{duration,dec}} = 100$  epochs until it reaches  $\beta_{\text{max}} = 10$ . 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  $\lambda_{\text{max}}$  and  $\beta_{\text{max}}$  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  $\beta_{\text{max}} = 50$  led to a significant degradation of reconstruction quality, indicating excessive distortion of the latent space.

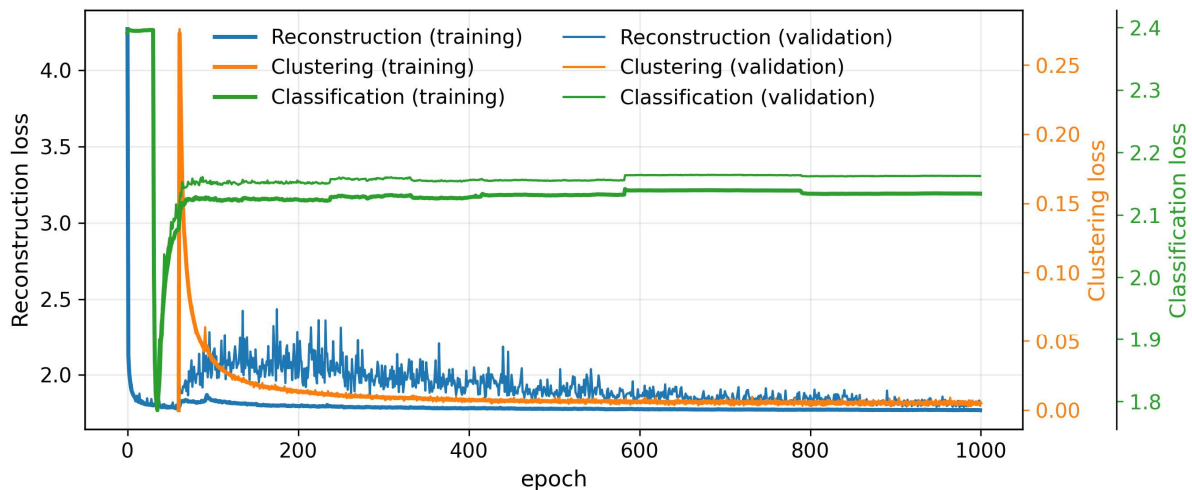


Figure 1 Training and validation evolution of the loss components: reconstruction loss  $L_{\text{rec}}$ , domain-classification loss  $L_{\text{domain}}$ , and clustering

---

### Comment 9

**Section 2.4: When you mention “a linear regression head”, does it refer to the last layer of the LSTM network, or to an independent model? In the latter, how is the LSTM trained?**

### Response

The linear regression head is a single fully connected layer applied to the LSTM output representation. The LSTM and this linear layer are trained end-to-end for DEM prediction, while the encoder remains fixed.

### **Change made in manuscript**

The following clarification was added verbatim to Section 2.4:

“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.”

---

### **Comment 10**

**Line 352: MLP is not introduced explicitly.**

### **Response**

Correct.

### **Change made in manuscript**

MLP is now explicitly defined as “multilayer perceptron” at first occurrence.

---

### **Comment 11**

**Section 2.6: Isn't two weeks a short period for testing, compared to one year for training? Is the autoencoder performance similar in all periods of the year?**

### **Response**

The training data does not correspond to a continuous full year. For training, we randomly sample 1,000 hours per turbine from 2023 to cover a wide range of operational conditions (totaling to 6 weeks of training). Evaluation of representation quality using MI and NMI is computationally intensive, which is why we restricted that analysis to two weeks of 2024 (temporally disjoint). For the fatigue task, we use a longer test period (June–September 2024) containing many start–stop events and broad operational variability.

### **Change made in manuscript**

We rewrote the dataset description. The following text is now included

For training, a random subset of 1,000 hours per turbine was selected from the year 2023, corresponding to approximately six weeks of data per turbine. Each turbine was assigned an anonymized identifier, and only 11 out of the 44 turbines were used for model training, corresponding to one quarter of the fleet. This split was adopted to limit overfitting and to explicitly assess generalization to unseen turbines. Model testing for operational-state inference was conducted on data from the first two weeks of 2024,

which constitute a temporally disjoint hold-out dataset used exclusively for evaluation and visualization. For the fatigue-related task, data from June to September 2024 were used for testing, as this period contains a high number of start–stop events and a wide range of operational conditions. We have reformulated the paragraph for better clarity

---

#### **Comment 12**

**Lines 488–490: “conservative lower bounds... model likely performs better”. Is this true? One could argue high-frequency data is more complex.**

#### **Response**

We agree the phrasing in the submitted version was misleading. Our intent was not to claim that the model “performs better” in an absolute sense, nor that higher-resolution SCADA would necessarily be easier to predict. The point is strictly about evaluation: SCADA variables are only available as 10-minute averages, so any intra-interval variability that may be captured by the vibration-based embeddings cannot be validated against the temporally averaged SCADA reference. This caps the measurable agreement with SCADA averages and should be interpreted carefully.

#### **Change made in manuscript**

We rephrased the statement to the following (your revised text, kept):

Because SCADA variables are available only as 10-minute 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 with respect to the unobserved instantaneous dynamics, rather than as an upper limit on achievable predictive performance.

---

#### **Comment 13**

**Line 500: If we perform post-hoc clustering, do we get clusters similar to DEC? Also ensure clustering loss does not negatively affect DEM estimation.**

#### **Response**

We now clarify this explicitly. Without DEC ( $\beta = 0$ ), the model yields three dominant groups: standstill, parked, and a single connected producing manifold. DEC mainly refines the producing manifold by partitioning it into multiple compact regimes, which makes post-hoc clustering easier and improves interpretability. With DEC enabled, post-hoc  $k$ -means clustering produces clusters similar to the DEC centroids.

We also acknowledge the trade-off: adding the DEC objective can slightly affect regression performance. A DEC-free model could maybe improve DEM regression, but the goal here is a unified SCADA-free framework for representation learning and regime identification; the DEM regression is included primarily as validation that fatigue-relevant information is preserved, and is not the main contribution of this paper, we kept the result as it is beating the current deployed model.

This behavior is also consistent with the loss curves: when DEC is introduced, the reconstruction loss temporarily increases and then recovers as the decoder adapts to the reorganized latent geometry.

### **Change made in manuscript**

- Added a figure showing the latent space without DEC .
- Added explicit discussion of DEC vs post-hoc clustering and the trade-off with regression.
- Included the following clarification in the Results/Regime section (your text, kept):

When no clustering constraint is applied (i.e., DEC is disabled and  $\beta = 0$ ), the 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 Fig2 two small clusters corresponding to non-producing conditions (standstill and parked), 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 Fig9 where producing states form a single connected manifold despite clear gradients in pitch angle.

By introducing DEC, the latent structure is 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 motivated by

interpretability and consistency with standard operational binning), the previously broad operating manifold is refined into multiple regimes that distinguish different levels of power production and control action. This five-cluster configuration aligns with the canonical division of turbine behavior used in SCADA-based classification, 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 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).

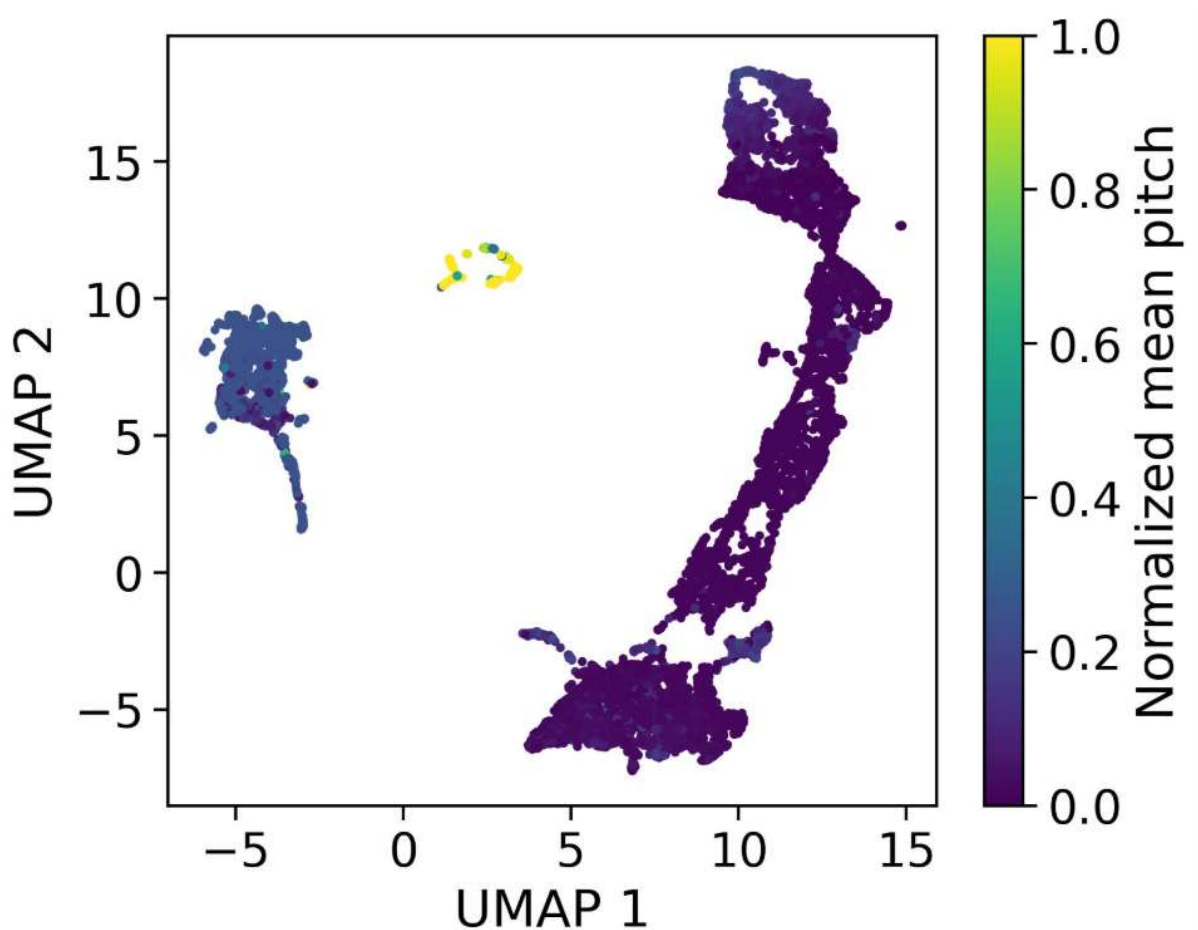


Figure 2. UMAP projection of the latent embeddings learned without Deep Embedded Clustering (DEC), coloured by normalised mean blade pitch. The representation separates parked and standstill conditions from operating states, but the operating regime remains a largely

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.

#### **Comment 14**

**Line 512: spelling mistake “there” instead of “their” (twice).**

#### **Response**

Correct.

#### **Change made in manuscript**

Both occurrences were corrected to “their”.

---

#### **Comment 15**

**What model is used for the SCADA baseline?**

#### **Response**

The SCADA baseline is a neural-network fatigue estimation framework previously developed and described in [1] and implemented by a co-author. We now state this explicitly and define the baseline in a dedicated subsection.

#### **Change made in manuscript**

A baseline-definition subsection was added:

Baseline definition:

In our contribution, we shall compare our approach to a pre-defined baseline. This reference baseline corresponds to an artificial neural-network-based fatigue estimation framework introduced in [1]. It combines standard 10-minute SCADA variables with acceleration metrics derived from simple statistical descriptors of vibration data e.g., RMS, standard deviation, etc.). This baseline can be viewed as the current state-of-the-art on farm-wide DEM estimation, and it is presently being deployed practically.

---

#### **Comment 16**

**Line 535: Why not using only SCADA as a baseline? If acceleration is included, then information used by the autoencoder+LSTM is partially included in the baseline. The comparison may evaluate architecture and processing more than SCADA vs acceleration.**

#### **Response**

We agree that a SCADA-only baseline (or an acceleration-only handcrafted baseline)

would be a more controlled information-parity comparator. However, our objective is not a sensor-ablation study. Instead, we benchmark against the current state-of-the-art solution used in practice for farm-wide DEM estimation, which combines SCADA variables with simple acceleration statistics.

We now state explicitly that the comparison is asymmetric by design. The baseline also has a strong advantage in training data volume (full year, and two years internally) compared to our 1,000 hours per turbine. The intent is therefore to position the proposed SCADA-free approach relative to a best-case practical reference.

We also added context that, in prior work on another wind farm, adding 10-minute acceleration statistics to SCADA-only models improved fatigue prediction performance, hence SCADA-only is not the strongest reference for this task.

### **Change made in manuscript**

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 [2]. It is therefore against this baseline that we benchmark our approach.

## **Response to Reviewer 2**

### **Response to Reviewer 2**

**We sincerely thank Reviewer 2 for the positive assessment of our work and for recommending acceptance after minor revisions. We appreciate the careful reading of the manuscript and the constructive comments, which helped us clarify methodological choices and improve the presentation. Each comment is addressed**

**below. All corrections and clarifications have been incorporated into the revised manuscript.**

---

### **Major Comment 1**

#### **Justification for using Deep Embedded Clustering (DEC)**

**Reviewer comment: The motivation for using DEC instead of simpler methods such as  $k$ -means is not sufficiently justified. Are the cluster boundaries stable when the number of requested clusters is varied?**

Response

We agree that a clearer justification was needed. DEC is not only used for clustering but its actually forcing the latent dimension to be clusterable, it is part of the training. If we rely only on K-means we would not be sure that the latent space would have clusters.

In the revised manuscript, we expanded the discussion in Section 3.1 to explicitly contrast the latent structure obtained without DEC and with DEC, and to explain why post-hoc clustering alone is insufficient in this context.

Changes made:

When no clustering constraint is applied (i.e., DEC is disabled and  $\beta=0$ ), the 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 Figure two small clusters corresponding to non-producing conditions (standstill and parked), 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 Fig3 where producing states form a single connected manifold the splitting is mainly driven by the pitch angle.

By introducing DEC, the latent structure is explicitly encouraged to become clusterable. DEC adds a set of learnable centroids 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 and control action. This five-cluster configuration aligns with the canonical division of turbine behavior used in SCADA-based classification, 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 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).

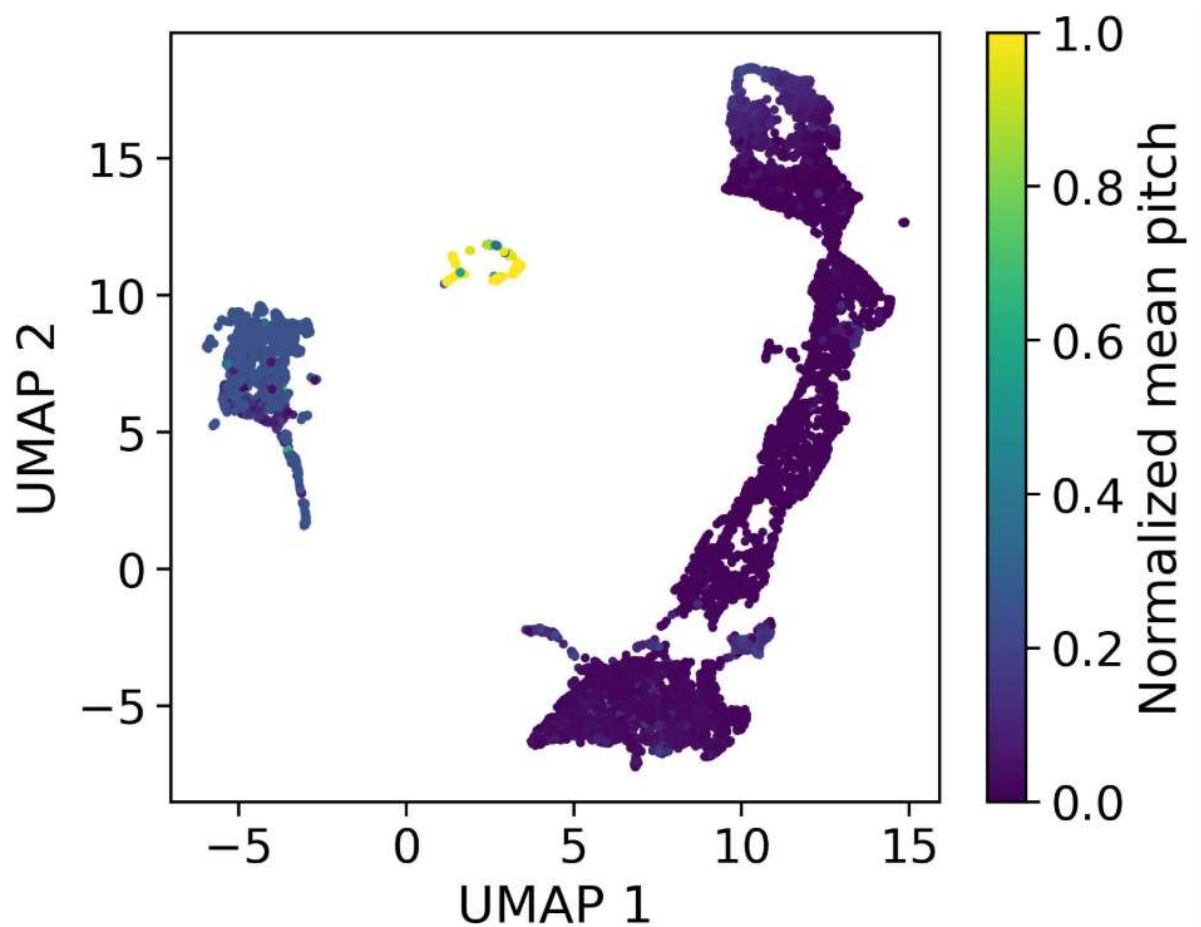


Figure 3 UMAP projection of the latent embeddings learned without Deep Embedded Clustering (DEC), coloured by normalised mean

*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.*

---

## Major Comment 2

Clarification of “train-on-4 / test-on-1” and fleet-wide generalization

Reviewer comment: The meaning of “train-on-4 / test-on-1” is unclear. Is this a split within

each fleet-leader turbine, or across turbines? This raises questions regarding fleet-wide deployment claims.

Response

We thank the reviewer for pointing out this ambiguity. The fatigue-estimation experiment follows a leave-one-fleet-leader-out strategy across the five strain-instrumented turbines. In the reported setup, the regression head is trained using DEM data from FL1–FL4 and evaluated on FL1–FL5. FL5 is entirely unseen during training. The encoder remains fixed during this stage.

This turbine-level split is intentional and necessary: DEM can only be computed on turbines equipped with strain gauges. As a result, metrics such as  $R^2$  and MSE cannot be evaluated on non-instrumented turbines, because no ground-truth fatigue reference exists for those units. Holding out one fleet-leader turbine is therefore the only practical way to evaluate generalization to unseen turbines while still allowing quantitative validation.

Change made in manuscript

We clarified the evaluation protocol in Section 3.4.2, and explicitly stated that FL5 is excluded from training and used only for evaluation. The following text was added (as in the revised manuscript):

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

The encoder is kept fixed, and only an LSTM-based regression head 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-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 over the full summer period, which includes a wide range of operational conditions and numerous stop–start events.

---

Major Comment 3

Effect of domain-adversarial training on DEM prediction

Reviewer comment: What is the DEM prediction performance with and without domain-adversarial training? Is there a risk of over-suppressing physically meaningful variability?

Response

The primary focus of this paper is operational-state inference and the learning of turbine-invariant operational embeddings. The DEM experiment is included as an auxiliary validation to demonstrate that the learned embeddings preserve load-related information, not as a dedicated study of fatigue modeling or of the interaction between domain adaptation and fatigue prediction.

For this reason, we did not perform a systematic ablation study of DEM prediction with and without domain-adversarial training. Nevertheless, the reported results indicate that domain-adversarial regularization does not prevent fatigue estimation: good predictive

performance is obtained on the held-out fleet leader FL5, while turbine identity is largely suppressed in the embeddings. This suggests that fatigue-relevant variability is retained despite suppressing turbine identity.

We agree that a focused study of the trade-off between turbine invariance and preservation of fatigue-relevant, site-specific variability (e.g., soil–structure interaction) is valuable and should be investigated in future work.

Change made in manuscript

We added an explicit statement in the Conclusions identifying this as future work. The revised manuscript now includes:

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.

---

Minor comments

Minor Comment 1

Lines 230–235: meaning of M vs. F

Response

We agree this was unclear.

Change made in manuscript

Clarified in Section architecture M now explicitly denotes the full input dimensionality and F the number of retained frequency bins.

---

Minor Comment 2

Line 420: “R2” typo

Response

Correct.

Change made in manuscript

Corrected to  $R^2$ .

---

Minor Comment 3

Line 500: missing “there is”

Response

Correct.

Change made in manuscript

Sentence rewritten to correct the missing copula.

---

Minor Comment 4

Line 540: paragraph ends suddenly

Response

We agree.

Change made in manuscript

Paragraph rewritten to answer reviewer 1

---

## References

[1] d N Santos, F., Noppe, N., Weijtjens, W., & Devriendt, C. (2021). Data-driven farm-wide fatigue estimation on jacket foundation OWTs for multiple SHM setups. *Wind Energy Science Discussions*, 2021, 1-36.

[2] de N Santos, Francisco, et al. "Farm-wide interface fatigue loads estimation: A data-driven approach based on accelerometers." *Wind Energy* 27.4 (2024): 321-340.