Response Letter

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 $t_{\mathrm{warm}}$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_warm_dann and t_warm_dec is heuristic, the rule was t_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 2.6** to clarify the role of t_{warm},
- 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 β. 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 \citep{kingma2014adam} with an initial learning rate of 5×10^{-3}. The learning rate was adapted using a \textit{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 50epochs were employed to ensure stable optimization.

Training followed a staged schedule under the composite objective in Eq.~\eqref{eq:total_loss}. First, the autoencoder was trained using only the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$for $t_{\mathrm{warm}} = 30$epochs to establish a stable reconstruction manifold. Second, the domain-adversarial loss $\mathcal{L}_{\mathrm{dom}}$was activated with a gradient reversal scale $\gamma = 0.4$, and its weighting coefficient $\lambda(t)$was linearly increased from $0$to $\lambda_{\max} = 0.8$over the subsequent 100epochs. Third, Deep Embedded Clustering (DEC) was introduced at epoch 60after initializing the cluster centroids using k-means with $n_{\mathrm{clusters}} = 5$. The clustering weight $\beta(t)$was then linearly ramped from $0$to $\beta_{\max} = 10$over 100epochs.

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 and 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 1000epochs, 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.

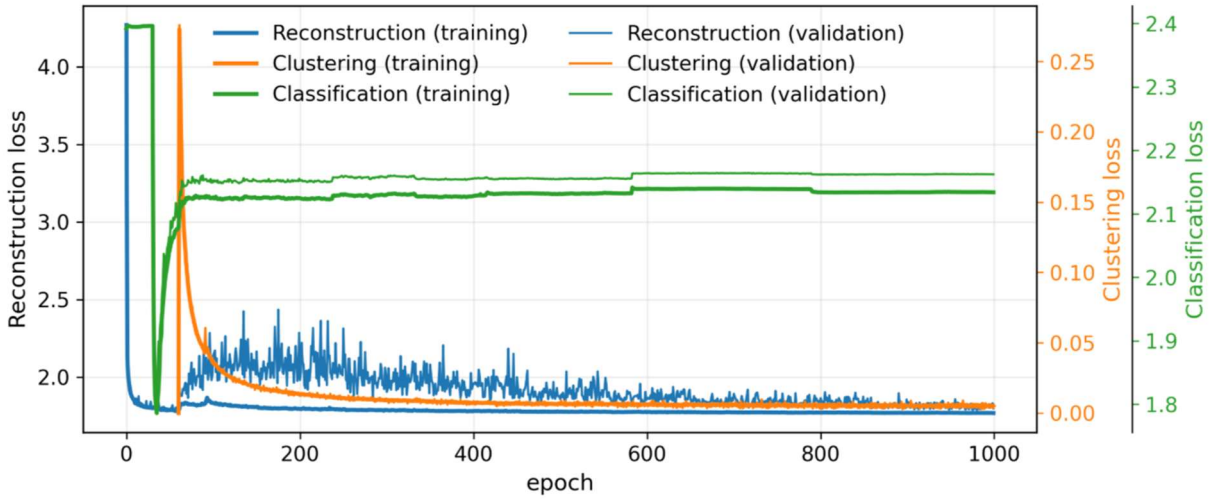

Figure 1 loss evolution through epochs

Change made in manuscript

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

Figure 1 summarizes the evolution of the loss components during training under the staged optimization strategy described in Section 2.7. During the warm-up phase (epochs 0--30), only the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$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 $t_{\mathrm{start,dann}} = 30$, the domain-adversarial objective is activated. Its weighting coefficient $\lambda$is increased linearly over $t_{\mathrm{duration,dann}} = 100$epochs until it reaches $\lambda_{\max} = 0.8$. Immediately after introduction, the domain-classification loss $\mathcal{L}_{\mathrm{domain}}$drops sharply. This transient behavior reflects the ability of the newly trained domain classifier $f_{\mathrm{dom}}$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.~\ref{fig:loss_curves}, 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 $t_{\mathrm{start,dec}} = 60$. Its weighting coefficient $\beta$is increased linearly over $t_{\mathrm{duration,dec}} = 100$epochs until it reaches $\beta_{\max} = 10$. Upon activation, the clustering loss $\mathcal{L}_{\mathrm{DEC}}$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.

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 \emph{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 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 Fig 2 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).

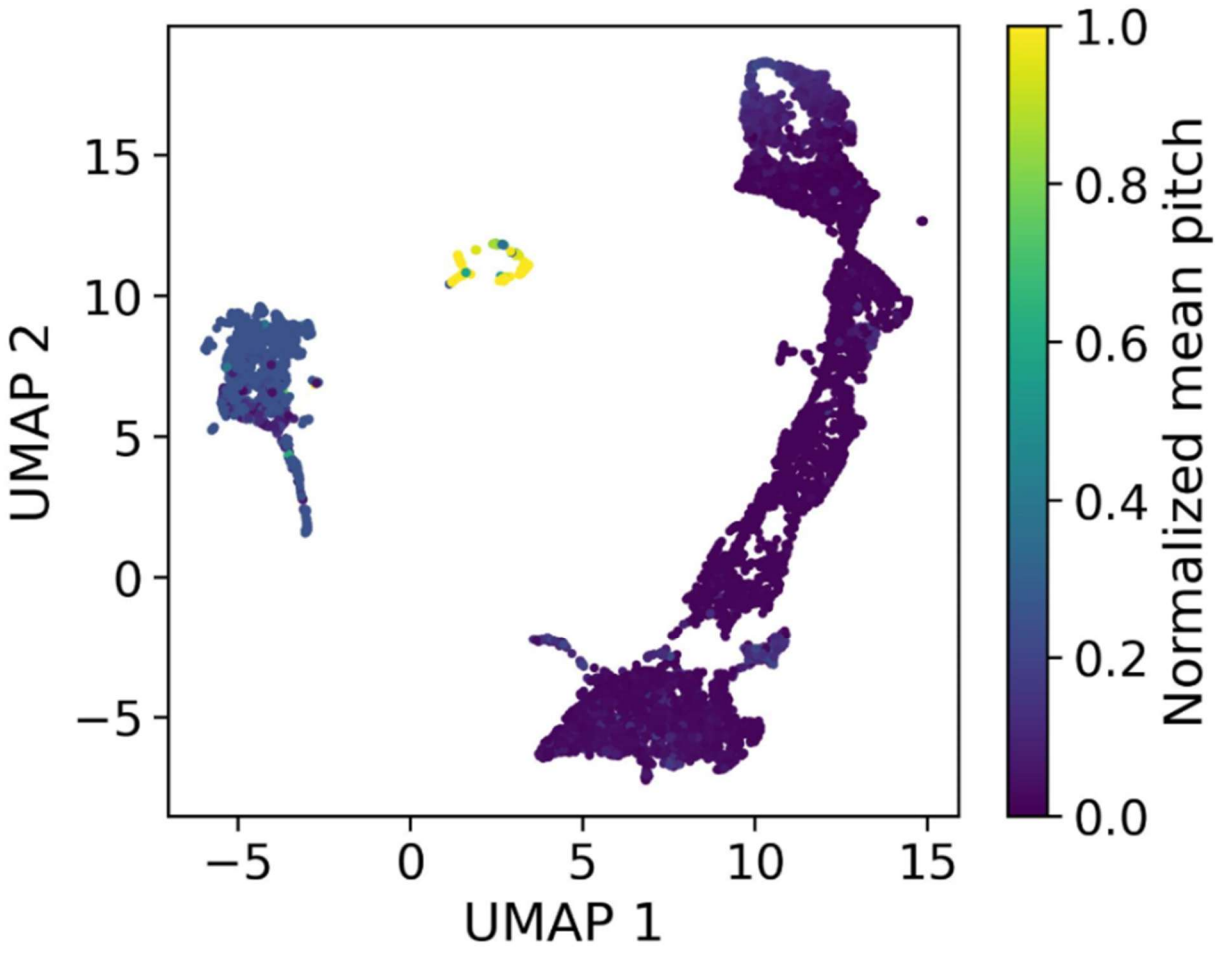

Figure 2 latent space without DEC

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:

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 \cite{de2024farm}. It combines standard 10-minute SCADA variables with acceleration metrics derived from simple statistical descriptors of vibration data (\textit{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.

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.