Working Paper - Learning Latent Geometries for Remaining Useful Life Prediction: A Geometry-Aware Variational Approach

Anirudh Krovi[1]

[1]PhD, Northwestern University; MBA, NYU Stern; Formerly at McKinsey & Company, anirudh.krovi@stern.nyu.edu

July 11, 2025

Abstract

Abstract

This paper introduces the FourierSpike-VAE, a novel geometry-aware variational autoencoder architecture for Remaining Useful Life (RUL) prediction. The model combines two complementary latent spaces: a smooth Fourier-based latent subspace designed to capture periodic or slowly varying degradation trends, and a spike latent subspace tailored to encode abrupt, transient anomalies. This hybrid decomposition enables the model to represent complex degradation dynamics with both fidelity and interpretability. We evaluate our approach on the NASA C-MAPSS FD001 turbofan engine dataset and show that the spike latents—though sparsely activated—carry rich semantic content aligned with end-of-life dynamics. Linear and nonlinear regressions from the spike space yield competitive forecasting performance, and t-SNE visualizations reveal a structured degradation manifold. Our findings highlight the promise of combining structured latent geometries with modular decoder pathways to model real-world time series where smooth trends and regime shifts coexist. We also discuss potential extensions to finance and healthcare domains.

1 Introduction

Variational Autoencoders (VAEs) [9] have emerged as a powerful framework for learning compressed, generative representations of complex datasets. By modeling the data-generating process through a latent variable structure, VAEs enable not only denoising and interpolation but also principled handling of uncertainty—making them a compelling choice for tasks ranging from image synthesis to sequential forecasting.

Recent research has highlighted the importance of the *geometry* of the latent space in VAEs. While early applications assumed Euclidean latent priors, later work explored non-Euclidean structures such as hyperspheres [4], Lie groups [6], and Fourier-inspired encodings [14, 13] to better capture symmetries, periodicities, and structural biases. These advances have proven critical in vision, rendering, and robotics, but their application to real-world forecasting tasks—particularly those involving shocks or degradation—remains limited.

In this work, we focus on a pressing real-world forecasting problem: predicting Remaining Useful Life (RUL) from degradation data. We use the C-MAPSS dataset [12], which simulates turbofan engine degradation under varying operational conditions. Each time series ends in system failure, making RUL prediction a form of time-to-event forecasting. Importantly, these degradation trajectories often contain abrupt transitions—akin to "shocks"—due to operational stress or component wear.

Traditional approaches to RUL modeling fall into two broad camps: (1) *Physics-informed models*, such as exponential degradation curves, which often struggle with heterogeneous regimes; and (2) *Machine learning* *models*, including CNNs, LSTMs, and attention-based models [16, 11, 15], which typically focus on temporal sequence architecture rather than representation geometry.

In parallel, recent work has shown that *Fourier feature encodings* and *periodic activations* can significantly improve a model's ability to capture high-frequency structure in space or time [14, 13]. However, these ideas have not yet been systematically tested in degradation modeling—especially with respect to how latent structure influences forecasting performance under sharp transitions or out-of-distribution (OOD) behavior.

Our prior experiments with toroidal latent spaces in MNIST and EMNIST showed that geometry-aware VAEs could model cross-domain "shocks" (e.g., digit-to-letter transitions) more gracefully than conventional VAEs. This motivated the current study: can similar insights be extended to real-world forecasting problems like RUL?

To this end, we propose a simple, geometry-aware VAE model: the encoder maps input time windows into a *Fourier-transformed latent space* (e.g., sinusoidal and cosine coordinates), and the decoder is a lightweight MLP trained to reconstruct inputs and estimate remaining life. Despite its simplicity, our model exhibits strong performance and interpretability—capturing degradation regimes and enabling smooth interpolation even near failure zones.

Our contributions are threefold: 1. We introduce *latent geometry* as a controllable inductive bias for time-to-event modeling in RUL tasks. 2. We show that even with simple decoders, structured latent spaces improve robustness to degradation shocks and OOD generalization. 3. We demonstrate conceptual and empirical links between degradation modeling and *financial regime prediction*, suggesting broader applicability to domains like credit risk and volatility forecasting [5, 8].

The remainder of this paper is organized as follows: Section 2 describes the NASA C-MAPSS dataset, outlines the FD001 subset used for our experiments, and details the preprocessing pipeline used to generate model-ready input windows. Section 3 introduces the FourierSpike-VAE architecture, explaining the rationale for its hybrid latent design, its encoder-decoder pathways, and the KL warm-up training strategy. Section **??** presents empirical results, including latent visualizations, RUL prediction performance, and regression analyses. Section **??** interprets the findings, discusses the role of geometry-aware encoding in capturing degradation dynamics, and outlines connections to forecasting problems in finance. Section **??** concludes by summarizing key takeaways, noting limitations, and proposing directions for future research.

2 Dataset and Preprocessing

2.1 Dataset Overview

We use the *C-MAPSS* dataset [12], developed by NASA's Prognostics Center of Excellence. It simulates degradation trajectories of turbofan engines under various operational conditions. Each engine unit is observed over multiple cycles (i.e., flights), with sensor and setting values recorded until system failure.

Each time step (row) in the dataset includes:

- `unit`: Engine identifier
- `cycle`: Time index (flight cycle)
- `setting1`, `setting2`, `setting3`: Operational conditions
- `s1` to `s21`: Sensor measurements

The goal is to predict the *Remaining Useful Life (RUL)* of an engine at each cycle, defined as the number of cycles remaining until failure.

2.2 FD001 Subset

C-MAPSS is partitioned into four subsets (FD001–FD004), each representing different combinations of operating conditions and fault modes. We focus exclusively on the *FD001* subset, which contains:

- One operating condition
- One fault mode

- 100 training units and 100 test units

This subset offers a controlled environment for testing geometry-aware latent representations before scaling to more complex settings.

2.3 Preprocessing Pipeline

We performed the following steps:

1. *Column definition and import*: The dataset was parsed using standard column names—three operational settings and twenty-one sensor signals.
2. *RUL computation*: For training units, RUL was computed as the difference between the unit's maximum cycle and its current cycle. For test units, RUL values were provided separately and aligned using unit indices.
3. *Normalization*: All sensor and setting columns were scaled to $[0, 1]$ using Min-Max normalization based on the training set. These statistics were then used to transform the test set.
4. *Sensor selection*: In our base configuration, we retain all 21 sensors and 3 settings, though later versions may explore dimensionality reduction or sensor importance ranking.

These preprocessing steps ensure a consistent, unit-independent representation of degradation, forming the foundation for modeling in subsequent sections.

3 Methodology

Our goal is to model degradation trajectories using a latent-variable generative framework that is geometry-aware and interpretable. We propose a hybrid Variational Autoencoder, which we refer to as the **FourierSpike-VAE**. This model combines a structured latent representation with a simple yet expressive decoding mechanism.

3.1 Training Patch Construction

To handle sequential time series data in a fixed-length format, we use a sliding window approach. Each training example consists of a window of $w = 30$ consecutive cycles of sensor and setting data, flattened into a single vector. The corresponding target is the RUL at the end of the window.

Given the preprocessed dataset, we group by engine unit and generate overlapping patches:

- Inputs: $x \in \mathbb{R}^{w \cdot d}$, where d is the number of features (21 sensors + 3 settings)
- Targets: $y \in \mathbb{R}$, the RUL value at the window's end

This patching step enables the model to learn degradation patterns from local history windows.

3.2 Rationale for Latent Geometry

The CMAPSS dataset exhibits a mixture of behaviors across engine trajectories: some features change smoothly with wear, while others signal abrupt shifts or localized anomalies indicative of mechanical faults. A unified model must therefore capture both:

- *Smooth degradation trends* — often captured by slow, periodic evolution in certain sensor readings (e.g., temperature, pressure).
- *Abrupt events or regime shifts* — such as sudden vibrations or threshold crossings, which are less predictable and highly localized in time.

Traditional VAEs may struggle to handle both types of signals within a single latent space. Our approach, the **FourierSpike-VAE**, introduces *inductive bias* through two complementary latent pathways:

- The *Fourier latent space* learns smooth, low-frequency components by decoding via a sinusoidal basis.
- The *Spike latent space* models sharper deviations using a flexible MLP decoder.

This design ensures that reconstruction is distributed meaningfully across two specialized components — without requiring hard-coded signal segmentation. In practice, the Fourier decoder captures baseline behavior, while the spike decoder activates selectively to correct for atypical or failure-prone behavior.

This architecture not only improves signal fidelity but also enhances interpretability by enabling downstream analyses of which latent dimensions correlate with impending failure. The clear separation of signal types allows for potential generalization to other domains (e.g., finance or healthcare) where similar smooth–shock duality exists.

3.3 Model Architecture

Based on the above discussion, **FourierSpike-VAE** is a hybrid variational autoencoder with two distinct latent representations:

1. A *Fourier latent space* that captures smooth, periodic patterns in degradation trajectories
2. A *Spike latent space* that captures more abrupt, non-smooth idiosyncrasies

The model operates on the flattened window x, and aims to reconstruct it via two decoders that specialize in complementary signal structures.

3.3.1 Encoders

Two separate encoder networks process the same input x to produce latent parameters:

$$\begin{aligned} \text{Encoder}_{\text{Fourier}}(x) &\to (\mu_f, \log \sigma_f^2) \in \mathbb{R}^{2z_f} \\ \text{Encoder}_{\text{Spike}}(x) &\to (\mu_s, \log \sigma_s^2) \in \mathbb{R}^{2z_s} \end{aligned}$$

These are used to sample latent variables using the standard VAE reparameterization trick:

$$z_f = \mu_f + \sigma_f \cdot \epsilon_f, \quad z_s = \mu_s + \sigma_s \cdot \epsilon_s \quad \text{with } \epsilon_f, \epsilon_s \sim \mathcal{N}(0, I)$$

3.3.2 Decoders and Latent Geometry

Each latent space has a dedicated decoder that reconstructs a signal over the window:

Fourier Decoder. The Fourier decoder treats $z_f \in \mathbb{R}^{2K}$ as amplitudes for a sinusoidal basis:

$$\hat{x}_{\text{Fourier}}(t) = \sum_{k=1}^{K} a_k \cos(2\pi k t) + b_k \sin(2\pi k t), \quad t \in [0, 1]$$

where a_k, b_k are extracted from the cosine and sine parts of z_f, and t is discretized over the input window[1] .

Spike Decoder. The spike decoder is a simple multilayer perceptron (MLP) that maps $z_s \in \mathbb{R}^{z_s}$ to the full reconstruction:

$$\hat{x}_{\text{Spike}} = \text{MLP}(z_s)$$

[1]The model does not directly optimize a_k and b_k as independent parameters. Rather, they are derived from the latent variable z_f, which is itself sampled from a distribution learned via the encoder. The Fourier decoder then interprets these components as sinusoidal amplitudes. What is optimized are the parameters of the encoder and decoder networks, which together learn how to produce effective a_k, b_k values to minimize reconstruction loss.

3.3.3 Reconstruction Flow

The final reconstruction is computed as:

$$\hat{x} = \hat{x}_{\text{Fourier}} + \hat{x}_{\text{Spike}}$$

This dual-path architecture allows the model to capture both globally smooth structures and local deviations in the input signal.

3.4 Latent Regularization and Loss

The training objective for the **FourierSpike-VAE** includes:

1. *Reconstruction Loss*:
$$\mathcal{L}_{\text{recon}} = \text{MSE}(x, \hat{x})$$

2. *KL Divergence* for each latent space:
$$\text{KL}_f = D_{\text{KL}}(q(z_f|x) \parallel \mathcal{N}(0, I)), \quad \text{KL}_s = D_{\text{KL}}(q(z_s|x) \parallel \mathcal{N}(0, I))$$
Computed analytically as:
$$\text{KL} = -\frac{1}{2}\sum(1 + \log\sigma^2 - \mu^2 - \sigma^2)$$

3. *Total Loss*:
$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \beta_f \cdot \text{KL}_f + \beta_s \cdot \text{KL}_s$$

We optionally vary the KL weights using a warm-up schedule, described next.

3.5 Training Procedure

The **FourierSpike-VAE** is trained using mini-batch stochastic gradient descent with the Adam optimizer (learning rate = 10^{-3}). Training is performed on GPU when available.

To ensure stable convergence and avoid latent collapse, we apply a *KL warm-up schedule*. But unlike standard practice, we use *opposing schedules* for the two latent spaces:

- The KL weight for the Fourier latent space, β_f, is increased linearly from 0.0 to 1.0 over the first 10 epochs.
- The KL weight for the spike latent space, β_s, is decreased linearly from 1.0 to 0.0 over the same period.

Formally:
$$\beta_f(t) = \min(1.0, 0.1 \cdot t), \quad \beta_s(t) = 1.0 - \beta_f(t)$$

This design encourages the model to first encode structure using the smooth Fourier basis and only introduce sharp features through the spike decoder later in training, once the base signal is well-modeled.

Each training step consists of:

- Forward pass: Encode x, sample z_f, z_s, reconstruct $\hat{x}$
- Compute loss using scheduled β_f, β_s
- Backward pass and parameter updates via Adam

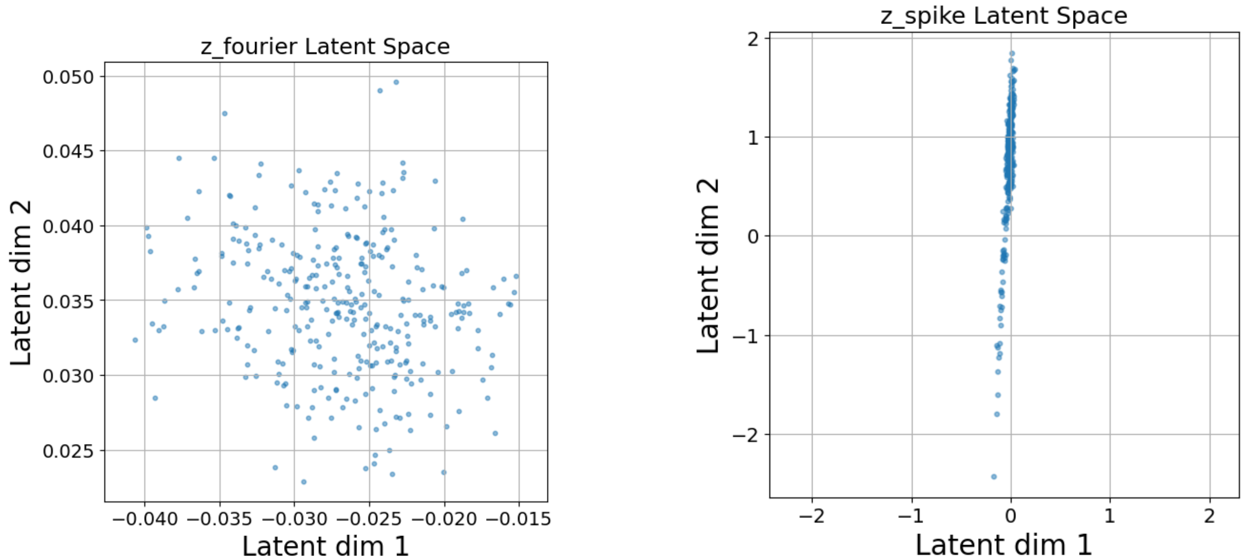
Training is typically run for 20–50 epochs depending on convergence and reconstruction quality.

4 Results and Interpretation

4.1 Latent Space Behavior

To understand how the **FourierSpike-VAE** decomposes signal structure, we visualize the learned latent spaces by extracting the encoder mean vectors for each input window. Both the Fourier and spike latent spaces are 8-dimensional, but for interpretability we plot only the first two dimensions of the mean vectors: $(\mu_{f,1}, \mu_{f,2})$ and $(\mu_{s,1}, \mu_{s,2})$.

Figure 1 shows the resulting 2D embeddings. Each point corresponds to a 30-cycle input patch, projected into its respective latent space.



(a) Fourier latent space z_f

(b) Spike latent space z_s

Figure 1: First two dimensions of the learned latent embeddings. The Fourier space exhibits smooth, isotropic clustering, while the spike space is highly concentrated, with most points collapsed near a single axis.

The contrast between these plots illustrates the model's structural disentanglement:

- The *Fourier latent space* z_f shows diffuse, circular spread—suggesting that the model encodes gradual variation (e.g., degradation patterns, operational regimes) in this space.
- The *spike latent space* z_s, in contrast, shows strong compression along one axis, with only one latent dimension significantly varying across examples. This indicates that spike-like corrections are rare and only activated when absolutely necessary.

This is a direct consequence of the opposing KL schedules: the model is initially encouraged to use the Fourier latent space (via increasing β_f), while being penalized for activating the spike space (via decreasing β_s). As a result, the spike decoder only comes into play for sharp or atypical deviations. Further, the figures suggest that a large portion of the Fourier latent space is necessary to capture gradual variations across samples, whereas only a narrow region of the spike latent space suffices to encode sharp deviations. The model treats spikes as specific and infrequent behaviors, aligning well with their semantic role in real-world degradation.

These results empirically support the central hypothesis of the **FourierSpike-VAE**: smooth components of degradation can be encoded geometrically via periodic bases, while idiosyncratic shocks are better handled via sparse, nonlinear adjustments. The model learns this structure naturally, without explicit supervision or spike labeling.

4.2 Spike Latents and RUL Prediction

We next evaluate whether the spike latent representation z_s encodes information about remaining useful life (RUL). While the **FourierSpike-VAE** is not trained to predict RUL directly, the learned latents may contain relevant structure. We fit a simple linear regression model from the spike latent space $z_s \in \mathbb{R}^8$ to the true RUL values on a held-out test set.

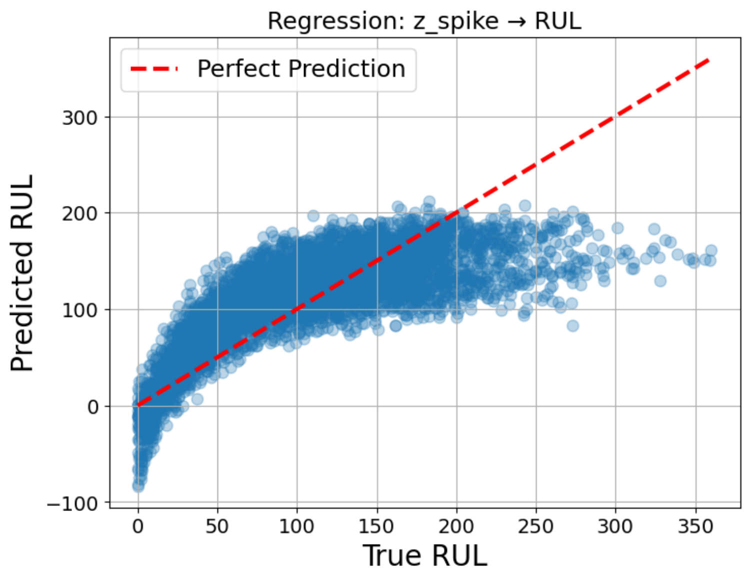


Figure 2: Linear regression from spike latents z_s to RUL. While the model underestimates high RUL values, it captures the overall decreasing trend.

The fit yields a mean absolute error (MAE) of 34.35 and an R^2 score of 0.564. As shown in Figure 2, the predicted RUL tracks the general trend, particularly in the lower-RUL regime where shock-like behavior is more prominent. The saturation at higher RUL values reflects the model's design: spike activations are suppressed early in an engine's life due to the reversed KL schedule, and only become informative as degradation progresses.

This result reinforces the interpretability of the spike latent space: although it is used sparingly (as seen in the compressed structure in Figure 1), it becomes increasingly predictive as failure approaches. The fact that a simple linear model can extract meaningful information from z_s confirms that the decoder has not only learned sparse representations, but also aligned them with real-world degradation semantics.

4.3 Nonlinear Regression: XGBoost on Spike Latents

To test whether the spike latent space z_s encodes more complex structure beyond what a linear model can extract, we fit a nonlinear gradient boosting model (XGBoost) from z_s to RUL. Compared to the linear fit, this model achieves improved performance: a mean absolute error (MAE) of 31.85 and an R^2 score of 0.587.

As seen in Figure 3, XGBoost captures additional curvature and interactions present in the latent space, especially in mid-RUL regions where degradation patterns become more variable. The improvement in predictive power supports the view that z_s encodes meaningful information about degradation, but does so in a nontrivial way that benefits from expressive modeling.

Together, these two regressions — one linear, one nonlinear — demonstrate that the spike latent space, though low-dimensional and tightly clustered, is not only interpretable but *predictively rich*. Its activation is rare, but when triggered, it aligns well with residual life estimation.

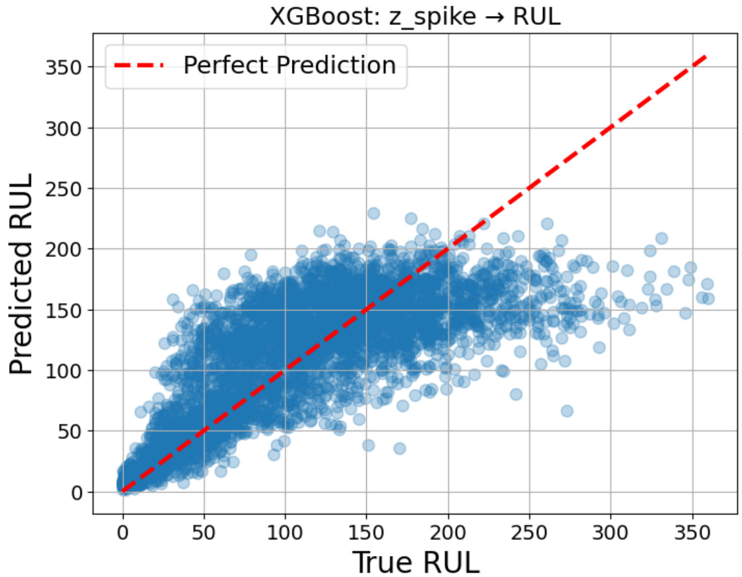


Figure 3: XGBoost regression from spike latent space z_s to RUL. Compared to the linear fit, this model shows better alignment in the mid-to-high RUL range.

4.4 Interpreting Latent Dimensions: $z_{\text{spike}}[0]$ Tracks Degradation Phase

To interpret individual spike dimensions, we plotted the first latent coordinate $z_{\text{spike}}[0]$ against the true RUL values. As shown in Figure 4, we observe a bell-shaped structure: values are low at the early and late stages of engine life, peaking in the mid-range.

This pattern implies that $z_{\text{spike}}[0]$ encodes something akin to a *window of failure emergence*, activating when early warning signs are detectable, and shutting off at both ends — when the system is either too healthy or too degraded to signal reliably.

This kind of latent interpretability is rare in black-box RUL models and highlights the power of the spike decoder in *specializing* to regimes that traditional methods may overlook.

4.5 Are Spike Latents Sufficient for RUL Prediction?

To test whether the spike latents alone capture the core signal for Remaining Useful Life (RUL), we trained an XGBoost regressor on the **combined** latent space $[z_{\text{spike}} \mid z_{\text{fourier}}]$. As shown in Figure 5, the resulting performance *did not significantly improve* over using only z_{spike} (see Table 1).

Table 1: RUL regression results across different latent spaces

Latent Input	**MAE**	R^2
z_{spike} (Linear)	34.35	0.564
z_{spike} (XGBoost)	31.85	0.587
$[z_{\text{spike}}, z_{\text{fourier}}]$ (XGBoost)	32.21	0.580

These results indicate that the *spike latents alone are sufficient* to drive downstream RUL estimation — highlighting their selective and compressed semantic content.

4.6 t-SNE Visualization of Latent Space Structure

To gain a more holistic view of how the latent variables encode degradation trajectories, we project the combined latent embeddings $(z_{\text{spike}}, z_{\text{fourier}})$ onto a 2D space using t-SNE. This nonlinear dimensionality

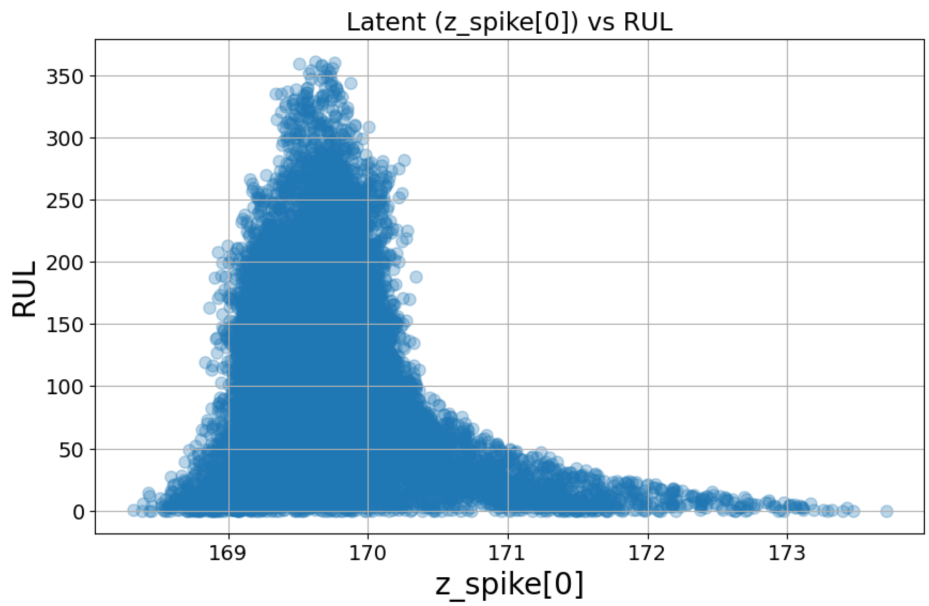


Figure 4: Relationship between the first spike latent dimension $z_{\text{spike}}[0]$ and RUL. The bell curve shape suggests that this coordinate activates during mid-life, when degradation signals are most interpretable.

reduction technique helps us understand the geometry of the learned latent space in a visually accessible way.

Figure 6 shows the resulting 2D embedding, where each point represents a data sample and is colored by its scaled Remaining Useful Life (RUL). Several patterns emerge:

- A smooth gradient in color suggests that latent representations form a continuous manifold aligned with the degradation process.
- Regions with low RUL (i.e., imminent failure) cluster distinctly, reinforcing the earlier result that z_{spike} captures meaningful end-of-life signals.
- The figure provides a global, nonlinear confirmation of the local linear regressions previously discussed.

This visualization supports our claim that the **FourierSpike-VAE** learns a structured latent geometry that aligns well with practical prognostics tasks such as Remaining Useful Life estimation.

5 Conclusion and Future Work

5.1 Core Idea

We introduced the **FourierSpike-VAE**, a novel generative model for time series data that disentangles smooth and abrupt latent structures. The architecture leverages two distinct encoding pathways: one for encoding continuous, periodic behavior via a sinusoidal basis ($z_{fourier}$) and another for capturing sharp, transient features through a more traditional multi-layer perceptron (z_{spike}). This design reflects a core hypothesis: time series in real-world systems—ranging from turbofan engines to financial markets—often contain both steady-state trends and sudden deviations, and each type of structure may be best represented with different latent geometries and decoders.

The Fourier pathway maps the latent vector $z_{fourier}$ to frequency components, reconstructing signals with smooth temporal variation. Meanwhile, the spike encoder-decoder path provides a high-resolution, nonlinear correction layer that becomes especially important near shocks, regime changes, or localized structural events. Together, these components provide both interpretability and adaptability in reconstruction and downstream tasks such as forecasting or anomaly detection.

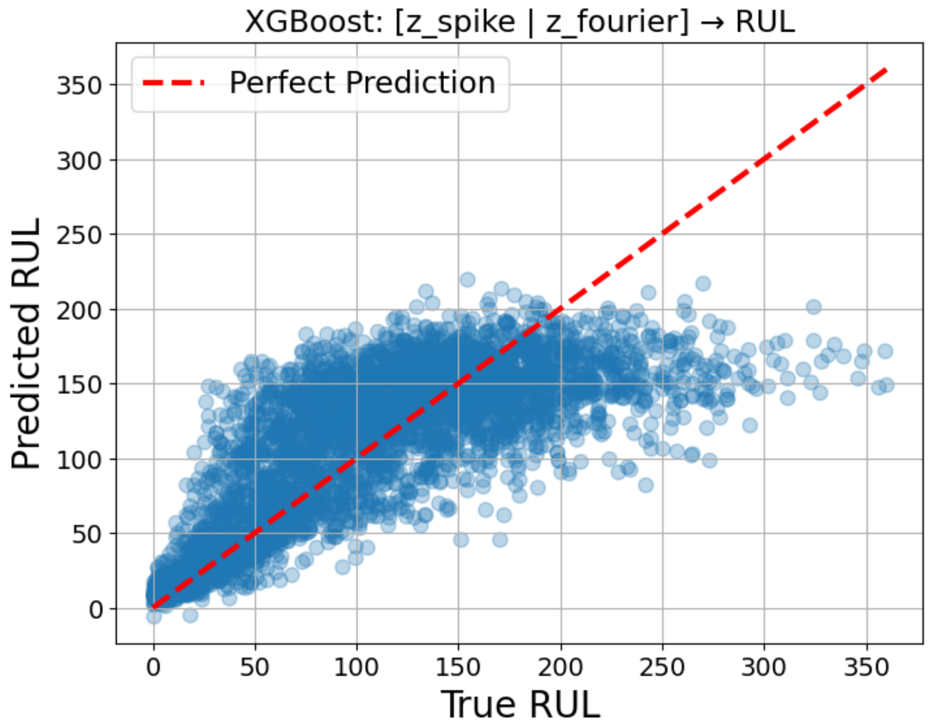


Figure 5: XGBoost regression using combined latent features $[z_{\text{spike}}, z_{\text{fourier}}]$. Compared to using z_{spike} alone, there is no meaningful gain in MAE or R^2, reinforcing the interpretability and focus of the spike latents.

5.2 Key Results

We evaluated our model on the NASA CMAPSS turbofan engine degradation dataset and observed the following:

- **Latent separation:** Visualization of the latent embeddings revealed that $z_{fourier}$ spans a broad, isotropic space, consistent with a role in capturing general structure. In contrast, z_{spike} was sparse and sharply concentrated, with meaningful activations clustered around degradation onset.
- **Predictive power of spikes:** A linear regression from z_{spike} to the Remaining Useful Life (RUL) target achieved an R^2 score of 0.564, with Mean Absolute Error (MAE) of 34.35. Using a nonlinear XGBoost regressor improved these scores to 0.587 and MAE of 31.85, respectively.
- **Minimal improvement with conditioning:** Conditioning on $z_{fourier}$ yielded no significant gain in predictive accuracy, suggesting that z_{spike} carries sufficient information for RUL prediction. This reinforces our hypothesis that degradation-related anomalies are localized and spike-dominant in nature.
- **Latent-RUL alignment:** Even a single latent dimension of z_{spike} showed strong alignment with RUL, indicating an interpretable degradation axis. Furthermore, t-SNE visualizations of the joint latent space showed a smooth gradient in RUL, suggesting that the model has learned a continuous degradation manifold.

5.3 Limitations

While the **FourierSpike-VAE** performs well in practice, there are several promising directions for improvement:

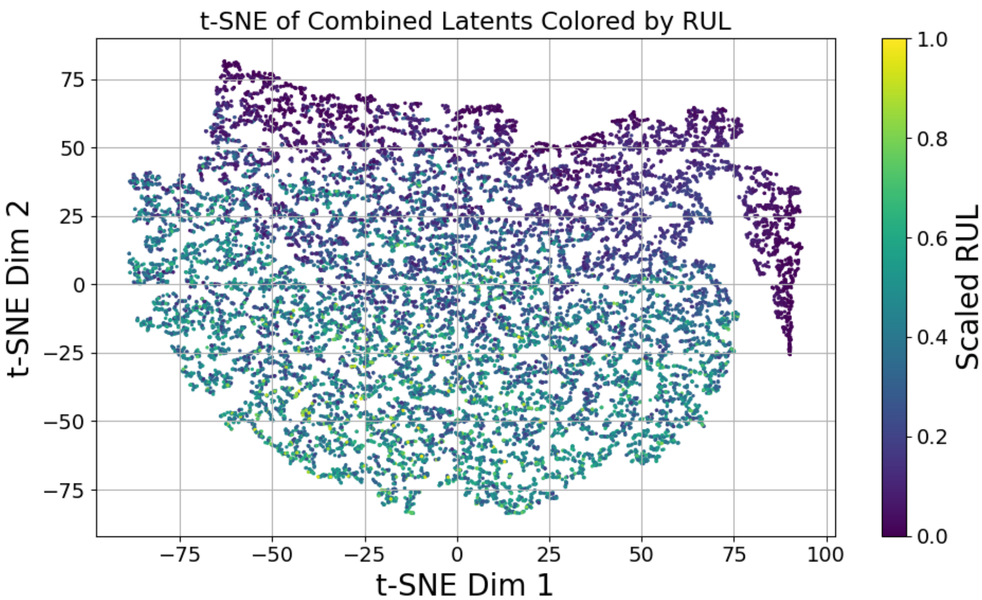


Figure 6: t-SNE projection of the combined latent embeddings ($z_{\text{spike}}, z_{\text{fourier}}$), colored by scaled Remaining Useful Life (RUL).

- **Latent disentanglement:** The model optimizes $z_{fourier}$ and z_{spike} jointly without explicit separation constraints. Incorporating disentanglement techniques—such as orthogonality penalties or information bottlenecks—could further clarify their roles.
- **Temporal priors:** The current model does not incorporate structured temporal priors (e.g., autoregressive dynamics, attention, or Gaussian processes). These could enhance modeling of long-term or multi-horizon behaviors.
- **Interpretability at scale:** While certain latent directions show alignment with degradation, more work is needed to systematically interpret or attribute meaning to individual dimensions, especially in high-dimensional settings.

5.4 Applications to Finance

The modeling framework proposed here is well-suited to financial applications where latent structures often blend smooth macro-trends with localized shocks. Possible extensions include:

- **Shock-aware latent forecasting:** Financial time series frequently experience abrupt changes due to earnings releases, policy changes, or macroeconomic events. A spike-aware latent system could improve robustness in forecasting such discontinuities.
- **Latent risk decomposition:** In factor modeling, distinguishing between latent ”trend factors” and ”shock factors” may aid in building interpretable, multi-timescale representations of risk.
- **Options and volatility modeling:** The impact of latent volatility spikes on options pricing is nontrivial. A hybrid VAE that isolates these spikes may provide better latent conditioning for volatility surfaces and regime-aware pricing.
- **Anomaly detection in trading or credit:** Spike latent activations could be treated as alarms for stress buildup or microstructural inefficiencies in financial systems.

5.5 Future Work

To extend the **FourierSpike-VAE** framework, we plan to explore the following directions:

- **Temporal priors:** Incorporate temporal smoothness or reversion priors over spike activations, especially via structured priors or latent diffusion processes [10, 3, 2, 1, 7].
- **Geometric latent spaces:** Extend from Gaussian latents to more expressive geometries such as toroidal or spherical embeddings for modeling cyclic variables.
- **Benchmarks in finance and healthcare:** Apply the architecture to finance (e.g., interest rate curves, volatility series) and healthcare (e.g., ECG data, clinical time series) to test generality.
- **Model interpretability tools:** Develop tools to understand latent attribution, e.g., sensitivity plots or latent saliency maps, especially around failure points or anomalies.

In summary, FourierSpike-VAE offers a modular and interpretable approach for modeling structured time series, opening avenues across domains where shocks and trends must be modeled together.

References

[1] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[2] Francesco Paolo Casale, Adrian V Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[3] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[4] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

[5] Jin-Chuan Duan, Irina Popova, and Peter Ritchken. Option pricing under regime switching. *Quantitative Finance*, 2(2):116–132, 2002.

[6] Luca Falorsi, Patrick de Haan, Tim R. Davidson, Nicola De Cao, Tim Falck, and Jakub M. Tomczak. Reparameterizing distributions on lie groups. *arXiv preprint arXiv:1903.02958*, 2019.

[7] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[8] Bryan Kelly and Seth Pruitt. Market expectations in the cross-section of present values. *The Journal of Finance*, 68(5):1721–1756, 2013.

[9] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[10] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[11] Xiaoyong Li, Qingyu Ding, and Jing-Qin Sun. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172:1–11, 2018.

[12] Abhinav Saxena and Kai Goebel. Turbofan engine degradation simulation data set. `https://www.nasa.gov/content/prognostics-center-of-excellence-data-set-repository`, 2008.

[13] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[14] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, et al. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] Cheng Zhang, Kun Ding, and Jing-Qin Sun. Multi-task learning with attention for rul prediction. *Neurocomputing*, 426:270–282, 2021.

[16] S. Zheng, K. Ristovski, R. Farahbakhsh, and R. Gautam. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7, 2017.