Self-Supervised Representation Learning for Remaining Useful Life Prediction: A Taxonomy and Triplet-Loss Case Study

Mohammad Badfar[1], Murat Yildirim[1], and Ratna Babu Chinnam[1]

[1] *Industrial and Systems Engineering, Wayne State University, Detroit, MI, 48201, U.S.*
mohammadbadfar@wayne.edu
murat@wayne.edu
ratna.chinnam@wayne.edu

ABSTRACT

Remaining useful life (RUL) prediction is fundamentally constrained by the scarcity of labeled run-to-failure data, motivating growing use of self-supervised learning (SSL) to exploit abundant unlabeled condition monitoring data. However, the literature lacks a systematic comparison of the loss functions underlying these SSL methods, as existing surveys organize the field by architecture rather than by the training objective itself. This paper reviews self-supervised loss functions for RUL prediction, spanning temporal-order, contrastive, metric-learning, multi-view, and meta-learning objectives, summarizing each family's assumptions, requirements, and limitations. To ground this review empirically, we present a framework pre-training a Transformer encoder with a triplet margin loss and evaluating it under few-shot fine-tuning on the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) turbofan benchmark. Across 2 to 50 labeled fine-tuning sequences, the pre-trained model consistently outperforms an identically structured supervised baseline, with the gap narrowing as labeled data becomes abundant, confirming that SSL pre-training is most valuable precisely in the label-scarce regime.

1. INTRODUCTION

In modern industrial systems, the acquisition of labeled condition monitoring data remains one of the most persistent and consequential challenges in predictive maintenance. Although sensor technology has advanced dramatically over the past decade, enabling continuous and high-resolution data collection from rotating machinery, turbofan engines, and other critical assets, the labels necessary for supervised learning, specifically run-to-failure trajectories annotated with remaining useful life (RUL) values, are inherently scarce. This scarcity arises from several compounding factors: allowing equipment to degrade to failure is economically prohibitive and operationally unsafe; failure events are statistically rare under modern maintenance regimes; and annotation of degradation stages requires expert domain knowledge that is costly to acquire. In aerospace, rail transport, and heavy manufacturing, the asymmetry between the abundance of unlabeled operational data and the scarcity of labeled failure data is extreme (B. Liu, Xu, Kang, Cao, & Zhao, 2025; Vermelin, Lövberg, & Kyprianidis, 2022; C. Yang et al., 2026).

RUL prediction has been studied under three broad methodological paradigms: physics-based modeling, statistical inference, and data-driven learning. Physics-based approaches construct explicit degradation models grounded in failure mechanics, such as Paris–Erdogan crack growth models or thermodynamic wear equations, that can produce interpretable predictions without historical data (Morales-Espejel & Gabelli, 2020; Paris & Erdogan, 1963). Statistical methods, including Wiener process models and Weibull distribution-based lifetime estimators, model the degradation process stochastically and offer tractable uncertainty quantification (Lai, Pan, Ong, & Chen, 2022), but impose strong assumptions of linearity or parametric form and are highly sensitive to parameter estimation under sparse data. Deep learning has since transformed RUL prediction methodology: convolutional networks demonstrated that raw sensor signals could be mapped directly to RUL labels without manual feature engineering (W. Zhang, Li, Peng, Chen, & Zhang, 2018; Zhao & Wang, 2021); recurrent architectures such as Long Short-Term Memory (LSTM) captured long-range temporal dependencies in degradation trajectories (Shi & Chehade, 2021; Faysal, Ngui, Lim, & Leong, 2023); and more recent architectures, including graph neural networks that model inter-sensor spatial dependencies (L. Wang, Cao, Xu, & Liu, 2022; J. Zhang et al., 2025), Transformer architectures that apply self-attention to capture global temporal patterns (Zhou & Wang, 2024), and hybrid decomposition–convolution frameworks (Wu, He, Wang, Ma, & Liu, 2025), have advanced the state of the art on benchmarks such as the Commercial Modular Aero-Propulsion

System Simulation (C-MAPSS) (Saxena, Goebel, Simon, & Eklund, 2008), the FEMTO-ST/PRONOSTIA bearing dataset (Nectoux et al., 2012), and the Xi'an Jiaotong University–Sun Yat-sen University (XJTU-SY) bearing dataset (B. Wang, Lei, Li, & Li, 2018). However, a critical limitation underlies virtually all of these methods: their performance is strongly conditioned on the availability of large labeled run-to-failure training sets, and when labeled data is reduced to a small fraction of full training corpora, generalization collapses (B. Liu et al., 2025; C. Yang et al., 2026).

This label scarcity problem has motivated a growing and increasingly diverse body of work applying self-supervised learning (SSL) and contrastive representation learning to RUL prediction, with the common goal of decoupling feature learning from the annotation bottleneck. What distinguishes these approaches from one another, however, is not the high-level SSL paradigm but the specific *loss function* used to shape the learned embedding space: the choice of pretext objective determines what notion of similarity the encoder is trained to preserve, what form of unlabeled supervision it requires (temporal order, augmentation invariance, cross-sensor co-occurrence, or explicit anchor–positive–negative structure), and ultimately how well the resulting representation transfers to few-shot RUL regression. Despite the rapid proliferation of SSL-based prognostic methods, the literature currently lacks a systematic comparison of the loss functions underlying these methods on common ground: existing surveys tend to organize the field by architecture or application domain rather than by the structural properties of the training objective itself, leaving practitioners without clear guidance on which loss family is appropriate for a given data regime, sensor modality, or degradation assumption.

A brief survey of the field illustrates the diversity of objectives already in use. Krokotsch et al. (Krokotsch, Knaak, & Gühmann, 2022) treat temporal proximity itself as the supervisory signal, training a network to regress the time difference between arbitrarily sampled windows. Closely related, the sequence-ordering framework of Vermelin et al. (Vermelin et al., 2022) instead poses a binary classification loss that distinguishes correctly ordered from shuffled window sequences on C-MAPSS. Both establish that temporal structure alone, without any RUL annotation, contains learnable degradation information, but both rely on a single-view autocorrelation-style loss that can limit representation discriminability when sensor channels behave similarly across distinct health states. Contrastive objectives address this limitation by explicitly contrasting multiple views or samples: the multisensor contrast neural network of Liu et al. (B. Liu et al., 2025) employs a cross-sensor similarity loss that maps vertical and horizontal acceleration spectrograms into a shared co-occurrence space; IAA-CRL (C. Yang et al., 2026) combines a differentiable, information-bottleneck-guided augmentation selector with multi-scale contrastive losses operating at both the instance and timestamp level; and DCSSL (Shen et al., 2026) enforces smoothness and monotonicity through a dual-dimensional contrastive loss built on dilated causal convolutions and temporal masking. Metric-learning losses based on triplet formulations offer a further alternative, directly enforcing relative-distance constraints between anchor, positive, and negative samples rather than a pairwise or batch-level contrastive term, and have seen use both offline and, in the case of OnlineTKS (Li, Bai, Liu, Chu, & Tan, 2025), in incremental online settings via soft triplet losses. Finally, meta-learning-based objectives, such as the Reptile-based attribution-consistency loss in MeMSAN (Ma & Zhang, 2026), depart from the pretext-task paradigm entirely, instead optimizing directly for rapid cross-task adaptation. Each of these loss families embodies a different inductive bias about what makes two degradation states "similar," and each carries distinct practical requirements and failure modes that have not, to date, been compared systematically within a single framework.

In this paper, we address this gap by providing a structured review of the self-supervised loss functions used for RUL prediction, organized around the formal properties of the objectives themselves rather than the architectures they happen to be paired with. For each loss family we summarize the underlying similarity assumption, the form of unlabeled supervision it requires, its principal strengths, and its known limitations, and we synthesize these comparisons into practical guidance on when each family is an appropriate choice. To ground this review empirically and to illustrate the review's taxonomy concretely, we additionally present a framework instantiating one loss family, the triplet margin loss, as a self-supervised pre-training objective for a Transformer encoder, and evaluate it under a controlled few-shot fine-tuning protocol on multiple publicly available benchmark datasets. The contributions of this paper are therefore threefold:

- A taxonomy of self-supervised loss functions currently used in RUL prediction and related prognostics tasks, organized by the structural form of the training objective;
- A comparative discussion of the strengths, requirements, and limitations of each loss family, including guidance on their applicability under different data regimes, sensor modalities, and degradation assumptions;
- A concrete framework applying a triplet-margin-loss instantiation of the metric-learning family to few-shot RUL prediction, validated against a supervised baseline on multiple benchmark datasets.

The remainder of the paper is organized as follows. Section 2 formalizes the few-shot RUL prediction problem shared across the reviewed methods. Section 3 presents the review of self-supervised loss functions. Section 4 describes the triplet-loss-based framework in detail, and Section 5 reports its experimental validation. Section 6 discusses the case study's

results in light of the broader review, and Section 7 concludes the paper.

2. PROBLEM FORMULATION

Before reviewing the landscape of self-supervised loss functions in Section 3, we formalize the few-shot remaining useful life (RUL) prediction problem that all of the reviewed methods, as well as the framework in Section 4, address. Establishing this shared formulation up front allows the loss functions discussed in Section 3 to be compared on common notational ground, since they differ primarily in the auxiliary objective used during pre-training rather than in the downstream prediction task itself.

Let an asset (e.g., a bearing or a turbofan engine) be monitored over its operational life through a multivariate sensor signal. At time step t, the raw sensor reading is denoted $x_t \in R^C$, where C is the number of sensor channels. A sliding window of length L ending at time t is defined as

$$X_t = [x_{t-L+1}, x_{t-L+2}, \ldots, x_t] \in R^{L \times C}, \tag{1}$$

which constitutes a single input sample to any of the models considered in this paper. The remaining useful life at time t, denoted y_t, is defined as the number of remaining operational cycles (or time units) before the asset reaches end-of-life (EOL), i.e., $y_T = 0$ at the failure cycle T. The overarching goal of RUL prediction, regardless of the pre-training strategy used to reach it, is to learn a mapping

$$f_\theta : R^{L \times C} \longrightarrow R, \qquad \hat{y}_t = f_\theta(X_t), \tag{2}$$

parameterized by θ, that accurately estimates y_t from the observed window X_t.

In the conventional supervised setting, θ is learned directly from a labeled dataset $\mathcal{D}_{\text{lab}} = \{(X^{(n)}_{1:T}, y^{(n)}_{1:T})\}_{n=1}^{N}$, where $X^{(n)}_{1:T}$ denotes the complete run-to-failure sensor trajectory of asset n and $y^{(n)}_{1:T}$ denotes the corresponding sequence of RUL labels. In practical industrial settings, however, only a small number of labeled run-to-failure trajectories are typically available, while a much larger pool of unlabeled condition monitoring data $\mathcal{D}_{\text{unlab}} = \{X^{(m)}_{1:T_m}\}_{m=1}^{M}$, collected from assets that have not necessarily reached failure or whose RUL annotations are unavailable, can be obtained at comparatively low cost. We refer to this regime, in which $N \ll M$, as the *few-shot RUL prediction* setting. This asymmetry between abundant unlabeled data and scarce labeled data is precisely what motivates the self-supervised loss functions reviewed in Section 3.

To address this regime, all of the methods considered in this review adopt some variant of a common two-stage paradigm. In the first stage, an encoder

$$g_\phi : R^{L \times C} \longrightarrow R^d, \tag{3}$$

is pre-trained in a self-supervised manner using only $\mathcal{D}_{\text{unlab}}$, without access to any RUL labels. Here, ϕ denotes the complete set of learnable parameters of the encoder, comprising the input projection weights W_{proj} and bias b_{proj} (Section 4.1), and the weights and biases of all N_{layer} Transformer encoder layers, including the query, key, and value projection matrices and feed-forward sublayer weights within each multi-head self-attention block. The objective of this stage is to learn a d-dimensional embedding space in which the relative position of an embedded window reflects the underlying, but unobserved, degradation state of the asset. It is precisely at this stage that the reviewed methods diverge: each defines a different self-supervised loss $\mathcal{L}_{\text{SSL}}(\phi; \mathcal{D}_{\text{unlab}})$ over the encoder parameters, encoding a different assumption about which unlabeled windows should be treated as similar or dissimilar in the embedding space. Section 3 reviews these choices of $\mathcal{L}_{\text{SSL}}$ in detail, and Section 4 instantiates one such choice, a triplet margin loss, as a concrete case study.

In the second stage, common to all reviewed methods, the pre-trained encoder g_ϕ is coupled with a lightweight regression head

$$h_\psi : R^d \longrightarrow R, \tag{4}$$

and fine-tuned on the scarce labeled set $\mathcal{D}_{\text{lab}}$, yielding the final predictive model $f_\theta = h_\psi \circ g_\phi$ as in (2). Here, ψ denotes the learnable parameters of the regression head, a small feed-forward network consisting of one or more fully connected layers that map the pooled d-dimensional embedding produced by g_ϕ to a scalar RUL estimate (the specific configuration used in our proposed method is given in Section 4.4). Depending on the method, the encoder parameters ϕ may either be frozen at their pre-trained values during this stage, isolating the contribution of the pre-trained representation, or allowed to adapt jointly with ψ; both variants appear across the methods surveyed in Section 3, and we adopt the frozen-encoder variant in our proposed framework (Section 4) to directly evaluate representation quality in isolation from further supervised adaptation.

The central hypothesis unifying all of the methods reviewed in this paper is that representations learned during self-supervised pre-training, regardless of the specific loss $\mathcal{L}_{\text{SSL}}$ used to learn them, capture degradation-relevant structure that substantially reduces the number of labeled samples required to achieve accurate RUL prediction, compared to training f_θ from randomly initialized weights on $\mathcal{D}_{\text{lab}}$ alone. Section 3 examines how different loss functions pursue this shared goal, and what trade-offs each entails.

3. REVIEW OF SELF-SUPERVISED LOSS FUNCTIONS FOR RUL AND PROGNOSTICS

3.1. Organizing Principle

Self-supervised learning (SSL) for remaining useful life (RUL) prediction spans a growing and increasingly heterogeneous set of methods. Rather than organizing this literature by network architecture or application domain, as prior surveys have largely done (R. Mo, Zhou, Yin, & Si, 2025), we organize it by the structural form of the self-supervised loss $\mathcal{L}_{\text{SSL}}$ introduced in Section 2: what notion of similarity the loss encourages the encoder to preserve, and what form of unlabeled supervision it requires to do so. This organizing principle exposes six loss families, reviewed in Sections 3.2–3.7: temporal-order and autoregressive pretext losses, contrastive instance-discrimination losses, metric-learning and triplet-based losses, reconstruction and masked-autoencoding losses, multi-view and cross-sensor contrastive losses, and meta-learning-based objectives. For each family, we summarize its underlying similarity assumption, the unlabeled supervision signal it requires, and its principal strengths and limitations for prognostic applications. Section 3.8 synthesizes these comparisons into a consolidated summary table and practical guidance for selecting a loss family under different data regimes and degradation assumptions. This taxonomy is complementary to, rather than a replacement for, existing few-shot-learning surveys in RUL prediction, which organize solutions by learning paradigm (transfer learning, meta-learning, data augmentation) rather than by loss function (R. Mo et al., 2025).

3.2. Temporal-Order and Autoregressive Pretext Losses

The earliest self-supervised objectives applied to RUL prediction treat the passage of time itself as the supervisory signal, requiring no augmentation design or explicit negative sampling. Krokotsch et al. (Krokotsch et al., 2022) pre-train a network to regress the time difference between arbitrarily sampled segments of an unlabeled run-to-failure trajectory, under the assumption that temporally proximate windows are more similar than temporally distant ones. In a related formulation, Vermelin et al. (Vermelin et al., 2022) pose the pretext task as binary classification, training the network to distinguish correctly ordered from randomly shuffled windows of C-MAPSS sensor data. Both methods establish that temporal structure alone, without any RUL annotation, contains learnable degradation information, and both report substantial reductions in labeled data requirements when the resulting representations are fine-tuned on a small labeled subset.

Strengths. These losses require only a single unlabeled trajectory per training instance, with no augmentation policy, negative-sampling design, or auxiliary architecture to tune, making them the simplest self-supervised objectives to implement.

Limitations. Because the objective is a purely single-view autocorrelation task, it can produce representations with limited discriminability across degradation stages that exhibit similar sensor behavior, particularly in early-to-mid life where degradation signatures are subtle (Krokotsch et al., 2022; Vermelin et al., 2022).

3.3. Contrastive Instance-Discrimination Losses

A broader family of losses generalizes the temporal-order idea into an explicit contrastive form, in which an encoder is trained so that two related ('positive') views of a sample are pulled together in embedding space while unrelated ('negative') samples are pushed apart. This family is best known through general-purpose time-series representation learning objectives such as TNC (Tonekaboni, Eytan, & Goldenberg, 2021), which defines temporal neighborhoods using local smoothness of the signal's generative process; T-Loss/SRL (Franceschi, Dieuleveut, & Jaggi, 2019), which treats random subseries of the same series as positives and subseries of other series as negatives; TS-TCC (Eldele et al., 2021), which contrasts weakly and strongly augmented views of the same window; and TS2Vec (Yue et al., 2022), which combines temporal and instance-wise contrastive losses in a hierarchical framework over multiple temporal resolutions.

Within RUL prediction specifically, several works instantiate this family directly. Kong et al. (Kong, Jin, Xu, & Chen, 2023) propose a contrastive learning framework that uses operational time as a proxy for degradation state, learning health indicators from unlabeled C-MAPSS trajectories in an unsupervised manner and subsequently regularizing the RUL regression stage to alleviate overfitting. Jang and Kim extend a Siamese-network contrastive formulation to learn a health-representation mapping for RUL prediction directly (Kong et al., 2023). Shen et al. (Shen et al., 2026) (DCSSL) combine temporal-level and instance-level contrastive losses with dilated causal convolutions and temporal masking to enforce smoothness and monotonicity constraints on bearing degradation representations. A closely related formulation, TACL-LSTM (Zhu, Lv, & Xie, 2025), constructs positive pairs via uniform random sampling across a bearing's full degradation cycle and negative pairs via temporally shuffled windows, explicitly selecting the resulting "trend contrast features" by their monotonicity and correlation with degradation state before downstream LSTM-based RUL regression. Song et al. (Song, Zheng, Lu, Liu, & Wang, 2026) propose a multi-task framework (IFSCL) in which a self-supervised contrastive module aligns raw input signals with high-dimensional latent features, preserving degradation-relevant information while jointly optimizing health-state assessment and RUL prediction on the XJTU-SY bearing dataset.

Strengths. Contrastive instance-discrimination losses are theoretically well studied, admit flexible choices of positive/negative

construction beyond pure temporal order (e.g., trend-based, feature-based), and have demonstrated strong empirical performance across both turbofan and bearing benchmarks.

Limitations. Performance is sensitive to the choice of augmentation or pairing strategy; augmentations borrowed from generic time-series or vision contrastive learning (e.g., random cropping, jittering) can violate physical degradation assumptions if applied without domain-specific adaptation, and most methods require careful tuning of temperature and batch-composition hyperparameters that are not always transferable across datasets (Yue et al., 2022; Zhu et al., 2025).

3.4. Metric-Learning and Triplet-Based Losses

A related but structurally distinct family replaces the batch-level contrastive objective with an explicit relative-distance constraint on a sampled triplet of anchor, positive, and negative windows. Rather than contrasting an anchor against many negatives simultaneously, the triplet margin loss enforces a fixed margin between the anchor-positive and anchor-negative distances, as detailed in our proposed framework in Section 4. Within RUL prediction, OnlineTKS (Li et al., 2025) applies a soft triplet loss in an online setting, combining self-supervised pre-training with incremental updates to accommodate distribution shift without negative transfer.

Strengths. Triplet-based losses require only a single negative per anchor (rather than a full batch of negatives), reducing memory and batch-size requirements relative to batch-contrastive losses, and their explicit distance-margin formulation gives a direct geometric interpretation of the learned embedding space that aligns naturally with the assumed monotonic degradation trajectory.

Limitations. Triplet losses are sensitive to the sampling strategy used to select positives and negatives (as discussed in Section 4.3); poorly chosen offsets can yield uninformative "easy" triplets that contribute negligible gradient signal, and, unlike batch-contrastive losses, triplet losses do not natively exploit multiple negatives per update, which can slow convergence relative to instance-discrimination objectives (Li et al., 2025).

3.5. Reconstruction and Masked-Autoencoding Losses

A fourth family dispenses with explicit positive/negative pair construction entirely, instead training the encoder to reconstruct masked or corrupted portions of its own input. Guo et al. (Guo et al., 2022) apply this idea directly to RUL prediction, pre-training a Transformer encoder-decoder with a gated convolutional unit to reconstruct randomly masked sensor patches from multi-sensor machine-tool data, in a formulation closely analogous to masked autoencoders in vision and language (He et al., 2022). In an earlier instantiation on the same C-MAPSS benchmark used in our case study, an autoencoder-based scheme combined with a generative adversarial double-error reconstruction objective is used to pre-train feature extractors that are subsequently fine-tuned with an LSTM for RUL regression (Hou, Xu, Zhou, Yang, & Fu, 2020). More broadly, general time-series masked-reconstruction frameworks such as PatchTST (Nie, Nguyen, Sinthong, & Kalagnanam, 2022), SimMTM (Dong et al., 2023), and TimesURL (J. Liu & Chen, 2024) (the latter combining reconstruction with a contrastive term) illustrate the maturity of this family outside prognostics specifically, and represent natural candidates for adaptation to RUL settings.

Strengths. Reconstruction-based losses require no explicit augmentation policy or negative sampling, scale straightforwardly to high-dimensional multi-sensor inputs, and directly leverage the full information content of the unlabeled signal rather than a derived similarity judgment.

Limitations. Because the objective optimizes fidelity to the raw input space, the encoder can expend representational capacity reconstructing sensor noise or task-irrelevant variation that has no bearing on degradation state, and reconstruction quality does not always correlate with downstream RUL prediction accuracy (Guo et al., 2022; Hou et al., 2020).

3.6. Multi-View and Cross-Sensor Contrastive Losses

A fifth family exploits the multi-channel structure of industrial condition monitoring data directly, contrasting representations across sensor channels or modalities rather than, or in addition to, across time. Liu et al. (B. Liu et al., 2025) propose a multisensor contrast neural network that maps vertical and horizontal acceleration spectrograms into a shared co-occurrence space with alternating contrastive objectives, outperforming simple channel-stacking baselines on FEMTO-ST. Yang et al. (C. Yang et al., 2026) (IAA-CRL) introduce a differentiable, information-bottleneck-guided augmentation selector that dynamically optimizes physics-constrained transformations, combined with multi-scale contrastive losses operating at both the instance and timestamp level. DCSSL (Shen et al., 2026), introduced above in Section 3.3, is dual-purpose in this respect: its temporal-level loss falls within the single-channel contrastive family, while its instance-level loss exploits cross-window, effectively cross-channel-analogous, comparisons.

Strengths. These methods directly exploit a structural property unique to industrial condition monitoring, namely the availability of multiple correlated sensor streams, yielding more discriminative representations than single-channel approaches on multi-sensor benchmarks.

Limitations. Their applicability is inherently constrained to settings with multiple correlated sensor channels; they do not transfer to single-channel monitoring scenarios, and the co-occurrence or augmentation-selection modules they rely on introduce additional architectural and tuning complexity

relative to single-view methods (B. Liu et al., 2025; C. Yang et al., 2026).

3.7. Meta-Learning-Based Objectives

A final family departs from the fixed-pretext-task paradigm shared by the previous five, instead optimizing directly for rapid adaptation across a distribution of related tasks. Model-agnostic meta-learning (MAML) is the predominant approach in this family within RUL prediction (Chang & Lin, 2025): Mo et al. (Y. Mo et al., 2023) apply MAML together with pseudo-meta-task construction based on time-sequence similarity, evaluated on C-MAPSS; MetaDESK (J. Yang et al., 2024) incorporates a Gaussian process into the MAML framework via variational inference to obtain sparse kernel features for few-shot RUL prediction; MetaDFKN (J. Yang & Wang, 2024) extends this direction with conditional normalizing flows and a lightweight contextual memory network for cross-domain few-shot RUL estimation; and a task-embedding variant of MAML (Shang et al., 2025) updates only low-dimensional task embeddings during adaptation, rather than the full model, to reduce overfitting risk under scarce labels. Ma and Zhang's MeMSAN (Ma & Zhang, 2026) instead adopts Reptile-based meta-learning combined with multi-scale spatial attention and attribution consistency objectives, demonstrating interpretable few-shot generalization on both C-MAPSS and bearing datasets. Bayesian extensions further calibrate the predictive uncertainty introduced by meta-learning under limited samples (Chang & Lin, 2025).

Strengths. Meta-learning objectives directly target the few-shot generalization goal, rather than relying on the "representation quality implies generalization" assumption underlying the other five families, and admit principled uncertainty quantification extensions.

Limitations. Nearly all meta-learning approaches require a distribution of diverse *labeled* source tasks for episodic meta-training, which is in tension with the unlabeled-data premise motivating the rest of this review; constructing meaningful pseudo-tasks from a single unlabeled trajectory remains an open challenge, and most reported results rely on task construction heuristics (e.g., time-sequence similarity, segmentation) whose sensitivity has not been extensively studied (Y. Mo et al., 2023; J. Yang et al., 2024).

3.8. Summary and Comparative Discussion

Table 1 summarizes the six reviewed loss families along the dimensions most relevant to practical adoption: the unlabeled supervision signal each requires, its principal degradation assumption, its relative sampling/design complexity, and representative datasets on which it has been validated.

Taken together, these comparisons suggest practical guidance for selecting a loss family under different conditions. When only a single sensor channel is available and the degradation process can be reasonably assumed monotonic, metric-learning losses such as the triplet margin loss offer a favorable balance of simplicity and representational structure, motivating their use in the case study of Section 4. When strong augmentation invariance is desired and computational budget permits larger batch sizes, contrastive instance-discrimination losses tend to yield the most discriminative representations, provided the augmentation policy is adapted to respect physical degradation constraints. When unlabeled data is abundant and sensor dimensionality is high, reconstruction-based losses scale favorably without requiring pair or triplet design. Multi-view contrastive losses are appropriate only when genuinely correlated multi-sensor data is available. Meta-learning objectives are best suited to settings where some labeled task diversity already exists across assets or operating conditions, despite their tension with the purely unlabeled-data premise of the other five families. Several open challenges persist across all six families, including the lack of a standardized few-shot evaluation protocol across datasets, limited theoretical understanding of why particular pretext tasks transfer to RUL regression specifically, and the absence of systematic ablations isolating the loss function from confounding architectural choices. Section 4 addresses one instance of this evaluation gap directly, by validating a triplet-margin-loss instantiation of the metric-learning family under a controlled few-shot protocol.

4. PROPOSED FRAMEWORK: TRIPLET-MARGIN-LOSS PRE-TRAINING FOR FEW-SHOT RUL PREDICTION

To ground the review in Section 3 with a concrete, empirically validated instantiation, this section presents a method applying the triplet margin loss, one member of the metric-learning family of self-supervised objectives, as the pre-training loss $\mathcal{L}_{\text{SSL}}$ introduced in Section 2. We first describe the shared Transformer encoder backbone (Section 4.1), then the triplet-based pre-training objective itself (Section 4.2), the sampling strategy used to construct triplets (Section 4.3), and finally the few-shot fine-tuning procedure used to adapt the pre-trained encoder to labeled RUL data (Section 4.4).

4.1. Transformer-Based Encoder Architecture

The backbone of the framework is a Transformer encoder g_ϕ, as introduced in (3), that maps an input window $X_t \in R^{L\times C}$ to a sequence of contextualized representations. This backbone is shared identically between the self-supervised pre-training stage (Section 4.2) and the fine-tuning stage (Section 4.4); only the lightweight task-specific heads attached on top of it differ between the two stages.

Table 1. Comparison of self-supervised loss families for RUL prediction.

Loss family	Supervision required	Core assumption	Design complexity	Representative datasets
Temporal-order / autoregressive	Single unlabeled trajectory, no augmentation	Temporal proximity ⇒ similarity	Low	C-MAPSS (Vermelin et al., 2022; Krokotsch et al., 2022)
Contrastive instance-discrimination	Augmented/paired views, negative batch	Augmentation invariance / trend consistency	Medium–High	C-MAPSS, FEMTO-ST, XJTU-SY (Kong et al., 2023; Shen et al., 2026; Zhu et al., 2025; Song et al., 2026)
Metric-learning / triplet	Anchor-positive-negative triplets	Proximate windows ⇒ similar degradation state	Medium	FEMTO-ST, PRONOSTIA (this work)
Reconstruction / masked autoencoding	Masking policy only, no pairs	Reconstructable structure ⇒ useful features	Low–Medium	C-MAPSS, machine-tool sensor data (Guo et al., 2022; Hou et al., 2020)
Multi-view / cross-sensor contrastive	Multiple correlated sensor channels	Cross-sensor co-occurrence reflects shared state	High	FEMTO-ST (B. Liu et al., 2025; C. Yang et al., 2026; Shen et al., 2026)
Meta-learning	Diverse *labeled* source tasks	Shared meta-knowledge transfers across tasks	High	C-MAPSS, bearing datasets (Ma & Zhang, 2026; Y. Mo, Li, Huang, & Li, 2023; J. Yang, Wang, & Luo, 2024)

4.1.1. Input Projection

Each window $X_t \in R^{L\times C}$ consists of L time steps with C raw sensor channels. A linear projection layer first maps each time step from the raw sensor space to the model's latent dimension d_{model}:

$$Z^{(0)} = X_t W_{\text{proj}} + b_{\text{proj}}, \qquad W_{\text{proj}} \in R^{C\times d_{\text{model}}}, \tag{5}$$

yielding $Z^{(0)} \in R^{L\times d_{\text{model}}}$.

4.1.2. Positional Encoding

Since the self-attention mechanism is permutation-invariant and the temporal ordering of sensor measurements is critical for capturing degradation progression, a positional encoding $P \in R^{L\times d_{\text{model}}}$ is added element-wise to the projected input:

$$Z = Z^{(0)} + P. \tag{6}$$

Each row of P encodes the position of a time step using fixed sine and cosine functions at varying frequencies across the d_{model} embedding dimensions. This encoding is fixed prior to training, not learned, and is therefore excluded from the encoder parameters ϕ.

4.1.3. Transformer Encoder Stack

The position-augmented representation Z is passed through a stack of N_{layer} Transformer encoder layers, each comprising multi-head self-attention with N_{head} heads followed by a position-wise feed-forward network, with residual connections and layer normalization:

$$E = \text{TransformerEncoder}(Z; N_{\text{layer}}, N_{\text{head}}), \tag{7}$$

where $E \in R^{L\times d_{\text{model}}}$ is the sequence of contextualized per-time-step embeddings, i.e., $g_\phi(X_t) = E$. This sequence representation serves as input to two distinct downstream heads: the pretext-task head used during self-supervised pre-training (Section 4.2), and the regression head used during few-shot fine-tuning (Section 4.4).

4.2. Self-Supervised Pre-Training via Triplet Loss

Among the loss families reviewed in Section 2, the triplet margin loss instantiates $\mathcal{L}_{\text{SSL}}$ as a metric-learning objective: the embedding space is shaped such that windows sampled from similar degradation states are pulled together, while windows sampled from dissimilar degradation states are pushed apart, using only the unlabeled set $\mathcal{D}_{\text{unlab}}$.

4.2.1. Embedding Extraction

For an input window X_t, the encoder produces a sequence of contextualized representations $E = g_\phi(X_t) \in R^{L\times d_{\text{model}}}$, as defined in (7). To obtain a single fixed-length embedding suitable for distance-based comparison, the sequence output is aggregated via mean pooling over the temporal dimension:

$$z = \frac{1}{L}\sum_{i=1}^{L} E_i \in R^{d_{\text{model}}}. \tag{8}$$

We denote this composite mapping from raw window to pooled embedding as $z = g_\phi(X_t)$ for notational convenience in the remainder of this subsection.

4.2.2. Triplet Formation

Each pretext-task instance is a triplet (X_a, X_p, X_n) consisting of an anchor window X_a, a positive window X_p assumed

to share a similar degradation state with the anchor, and a negative window X_n assumed to originate from a dissimilar degradation state. The precise criteria used to select positive and negative samples relative to a given anchor are detailed in Section 4.3. Each element of the triplet is passed independently through the shared encoder to obtain the corresponding embeddings:

$$z_a = g_\phi(X_a), \quad z_p = g_\phi(X_p), \quad z_n = g_\phi(X_n). \tag{9}$$

4.2.3. Triplet Margin Loss

The encoder is optimized to minimize the Euclidean distance between the anchor and the positive embedding while maximizing the Euclidean distance between the anchor and the negative embedding, subject to a margin α. Formally, with

$$d_{ap} = \|z_a - z_p\|_2, \qquad d_{an} = \|z_a - z_n\|_2, \tag{10}$$

the triplet loss for a single triplet is given by

$$\mathcal{L}_{\text{triplet}} = \max(0,\, d_{ap} - d_{an} + \alpha), \tag{11}$$

The margin α specifies the minimum amount by which d_{an} must exceed d_{ap} for the loss to be zero, preventing the encoder from collapsing all embeddings to a single point and controlling how aggressively dissimilar degradation states are pushed apart in the embedding space. The encoder parameters ϕ are updated by minimizing the expected loss over all sampled triplets,

$$\phi^\star = \arg\min_\phi \; E_{(X_a, X_p, X_n)\sim\mathcal{D}_{\text{unlab}}} \left[\mathcal{L}_{\text{triplet}}\right]. \tag{12}$$

Intuitively, minimizing (11) encourages the encoder to organize the embedding space according to the underlying, but unobserved, degradation progression of the monitored asset, since temporally and physically proximate degradation states are repeatedly drawn as positive pairs while distant degradation states are drawn as negative pairs. This structure is subsequently exploited during fine-tuning (Section 4.4), where only a small number of labeled samples are needed to map the pre-organized embedding space to absolute RUL values. Section 2.3 situates this objective relative to other metric-learning and contrastive alternatives.

4.3. Triplet Sampling Strategy

The effectiveness of the triplet loss in (11) depends critically on how positive and negative windows are selected relative to a given anchor. We adopt a temporal-proximity sampling strategy grounded in the assumption that the degradation state of an asset evolves gradually and monotonically over its operational life: windows that are close in time are likely to reflect similar degradation conditions, whereas windows separated by a sufficiently large time gap are likely to reflect substantially different degradation conditions. All anchor, positive, and negative windows are drawn from the same unit trajectory; no cross-unit sampling is performed, in order to avoid conflating inter-unit variability with intra-unit degradation progression.

4.3.1. Anchor Selection

For each training iteration, an anchor window X_a is sampled uniformly at random from the set of valid window start indices within a given unit's unlabeled trajectory.

4.3.2. Positive Selection

Given an anchor at time index t_a, a positive window X_p is sampled from the same trajectory at a time index t_p satisfying

$$|t_p - t_a| \le w_{\text{pos}}, \tag{13}$$

where w_{pos} denotes the maximum allowable temporal offset for a positive sample. This small window enforces that positive pairs originate from approximately the same degradation state, providing a self-supervised analogue to label similarity in the absence of ground-truth RUL annotations.

4.3.3. Negative Selection

Given the same anchor at time index t_a, a negative window X_n is sampled from a later point in the same trajectory, at a time index t_n satisfying

$$t_a + g_{\min} \le t_n \le t_a + g_{\max}, \tag{14}$$

where $g_{\min}$ and $g_{\max}$ define the minimum and maximum temporal offsets, respectively, for a negative sample. Restricting the negative offset to this bounded range, rather than sampling from the entire remainder of the trajectory, ensures that negatives consistently represent a degradation state meaningfully distinct from the anchor, while remaining within the same unit and thus avoiding the confound of unit-to-unit variability. Sampling negatives exclusively later in the trajectory reflects the irreversible and monotonic nature of mechanical degradation: a window drawn between $g_{\min}$ and $g_{\max}$ steps after the anchor is, by construction, closer to end-of-life and therefore corresponds to a more advanced degradation stage than the anchor.

Together, (13) and (14) define the triplet sampling distribution from which (X_a, X_p, X_n) triplets are drawn during pre-training (Section 4.2). Since the characteristic degradation timescale differs across datasets, the values of w_{pos}, $g_{\min}$, and $g_{\max}$ are tuned separately for each dataset and are reported in Section 5.

4.4. Fine-Tuning for Few-Shot RUL Prediction

Following self-supervised pre-training (Section 4.2), the encoder g_ϕ is adapted to the downstream RUL prediction task using only the scarce labeled set $\mathcal{D}_{\text{lab}}$. This stage introduces a lightweight regression head on top of the pre-trained encoder,

kept intentionally low-capacity relative to the encoder so that only a small number of parameters must be estimated from the few labeled samples available, and trains it to map the learned embedding space to absolute RUL values.

4.4.1. Regression Head

For a labeled window X_t with associated RUL label y_t, the encoder first produces the contextualized sequence representation $E = g_\phi(X_t) \in R^{L \times d_{\text{model}}}$, as in (7), which is then mean-pooled over the temporal dimension to obtain a fixed-length embedding $z = g_\phi(X_t) \in R^{d_{\text{model}}}$, identical to the pooling operation used during pre-training (8). A regression head h_ψ, consisting of a small feed-forward network, maps this embedding to a scalar RUL prediction:

$$\hat{y}_t = h_\psi(z) = h_\psi\big(g_\phi(X_t)\big). \tag{15}$$

Using a shared pooling operation across both stages ensures that the embedding space organized during triplet pre-training is consumed by the regression head in the same form in which it was optimized.

4.4.2. Frozen Encoder Fine-Tuning

To directly evaluate the quality of the representations learned during self-supervised pre-training, the encoder parameters ϕ are kept frozen at their pre-trained values throughout fine-tuning; only the regression head parameters ψ are updated. This design choice isolates the contribution of the pre-trained embedding space: since g_ϕ is not permitted to adapt to the labeled data, any improvement in RUL prediction accuracy can be attributed to the structure already present in the embedding space as a result of triplet-loss pre-training, rather than to additional encoder capacity exposed to labeled samples. We note that this frozen-encoder choice is itself a design axis distinguishing methods surveyed in Section 3, several of which instead allow joint adaptation of ϕ and ψ.

4.4.3. Fine-Tuning Objective

The regression head is trained to minimize the mean squared error (MSE) between predicted and ground-truth RUL values over the labeled set $\mathcal{D}_{\text{lab}}$:

$$\psi^\star = \arg\min_{\psi} \frac{1}{|\mathcal{D}_{\text{lab}}|} \sum_{(X_t, y_t) \in \mathcal{D}_{\text{lab}}} \big(h_\psi(g_{\phi^\star}(X_t)) - y_t\big)^2, \tag{16}$$

where $\phi^\star$ denotes the frozen, pre-trained encoder parameters obtained from (12). Because $|\mathcal{D}_{\text{lab}}|$ is small by construction in the few-shot setting (Section 2), only the comparatively low-capacity head h_ψ needs to be fit to the labeled data, substantially reducing the risk of overfitting relative to fine-tuning or training the full encoder-head pipeline from scratch. Section 5 reports the experimental validation of this framework across multiple benchmark datasets.

5. EXPERIMENTAL VALIDATION

This section reports the experimental validation of the triplet-loss method introduced in Section 4. We evaluate the extent to which self-supervised pre-training improves few-shot RUL prediction accuracy relative to a supervised baseline trained from scratch, under an identical Transformer architecture and an identical fine-tuning protocol, varying only the presence or absence of triplet-loss pre-training.

5.1. Dataset

We evaluate the proposed framework on the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset (Saxena et al., 2008), a widely used benchmark for RUL prediction of turbofan engines. C-MAPSS consists of multivariate run-to-failure sensor trajectories generated from a thermodynamic simulation of a turbofan engine, spanning multiple operating conditions and fault modes across four sub-datasets (FD001–FD004), each comprising a training set of complete run-to-failure trajectories and a test set of trajectories truncated prior to failure. In this study, we use the FD001 sub-dataset for all pre-training, fine-tuning, and final evaluation experiments. Each trajectory records 21 sensor channels (e.g., total temperature at fan inlet, physical fan speed, static pressure at high-pressure compressor outlet) together with three operational settings, sampled once per operational cycle. In the training set, each engine unit runs from a healthy initial state until failure, so the RUL label at every cycle can be computed directly as the number of cycles remaining until the end of the trajectory. In the test set, each unit's trajectory is truncated at a randomly chosen cycle prior to failure, and the ground-truth RUL at the truncation point is provided separately for evaluation. Following common practice, RUL labels are clipped at a maximum value to reflect the assumption that degradation is not yet detectable during the early healthy operating period, and sensor channels are normalized per-channel prior to windowing. In this case study, the C-MAPSS training set is used in two distinct roles: first, in its entirety and without RUL labels, as the unlabeled pool $\mathcal{D}_{\text{unlab}}$ for triplet-loss pre-training (Section 4.2); and second, subsampled into few-shot subsets, as the labeled pool $\mathcal{D}_{\text{lab}}$ for fine-tuning. The official C-MAPSS test set is reserved exclusively for final evaluation and is not used at any stage of pre-training, fine-tuning, or model selection.

5.2. Experimental Setup

Table 2 summarizes the encoder, triplet-loss, and fine-tuning hyperparameters used in all experiments reported below.

5.2.1. Pre-Training

The Transformer encoder described in Section 4.1 is pre-trained using the triplet margin loss (Section 4.2) on the unlabeled C-MAPSS training trajectories, with triplets sampled

Table 2. Hyperparameters used for the encoder, triplet-loss pre-training, and fine-tuning stages.

Parameter	Value
Input window	
Sequence length L	40
Transformer encoder	
Model dimension d_{model}	64
Number of layers N_{layer}	1
Number of attention heads N_{head}	8
Dropout	0.1
Triplet sampling and loss	
Margin α	0.2
Positive window w_{pos}	5
Negative offset range $[g_{\min}, g_{\max}]$	[10, 30]
Triplets sampled per unit	100
Pre-training	
Optimizer	Adam
Batch size	32
Epochs	100
Fine-tuning / regression head	
Regression head layers	1
Optimizer	Adam
Batch size	32
Epochs	100
Evaluation	
Random seeds / repeats per shot count	10

according to the temporal-proximity strategy of Section 4.3. No RUL labels are used at this stage.

5.2.2. Few-Shot Fine-Tuning Protocol

To evaluate the effect of pre-training under label scarcity, the pre-trained encoder is fine-tuned on labeled subsets of the C-MAPSS training set of increasing size, following the frozen-encoder procedure of Section 4.4. We define a "shot" as a single labeled input sequence (i.e., one windowed sample X_t paired with its RUL label y_t), and evaluate fine-tuning set sizes of $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50\}$ shots, in addition to a *full-data* setting in which the encoder is fine-tuned on the entire labeled training set. For each shot count below the full-data setting, labeled samples are drawn at random from the training set, and results are evaluated on the complete, held-out C-MAPSS test set.

5.2.3. Baseline: Supervised Training Without Pre-Training

To isolate the contribution of triplet-loss pre-training, we train a baseline model using an architecturally identical Transformer encoder and regression head, with weights randomly initialized rather than initialized from the pre-trained encoder $g_{\phi^\star}$. This baseline is trained end-to-end (i.e., without freezing any layers) directly on the same few-shot labeled subsets and the same full training set as the proposed method, using the same MSE objective (16). Comparing this baseline against the proposed method under identical shot counts directly measures the effect of self-supervised pre-training, independent of any differences in architecture, optimizer, or data split.

5.2.4. Evaluation Metrics

Model performance is evaluated on the official C-MAPSS test set using mean squared error (MSE) and mean absolute error (MAE) between predicted and ground-truth RUL values.

5.3. Results

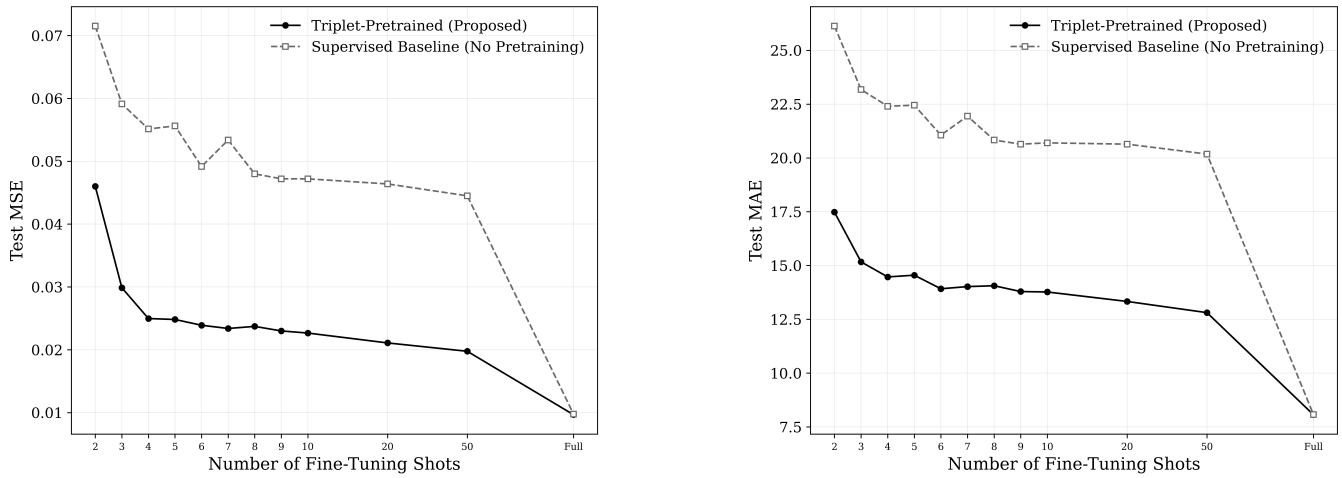
Figures 1a and 1b report the test-set MSE and MAE, respectively, of the proposed triplet-pretrained model and the non-pretrained baseline across all evaluated shot counts.

As shown in Figure 1a and Figure 1b, the triplet-pretrained model achieves substantially lower MSE and MAE than the non-pretrained baseline across the entire low-shot regime (2–50 shots). The performance gap between the two methods is largest at the smallest shot counts, where the baseline model, lacking any pre-trained structure in its embedding space, struggles to fit a meaningful mapping from only a handful of labeled sequences. As the number of fine-tuning shots increases toward 50, the gap between the two methods narrows, consistent with the baseline gradually compensating for the absence of pre-training as more labeled supervision becomes available. In the full-data fine-tuning setting, the two methods converge to comparable MSE and MAE, indicating that the advantage conferred by triplet-loss pre-training is concentrated in the label-scarce regime and diminishes once sufficient labeled data is available to train the encoder-head pipeline end-to-end from scratch.

These results support the central hypothesis introduced in Section 2: representations learned via triplet-loss self-supervised pre-training capture degradation-relevant structure that substantially reduces the number of labeled samples required for accurate RUL prediction, without providing additional benefit once labeled data ceases to be the limiting factor. Section 6 discusses these findings in the context of the broader review of self-supervised loss functions presented in Section 3.

6. DISCUSSION: CONNECTING THE CASE STUDY TO THE BROADER REVIEW

Sections 3 and 4 were deliberately structured so that the triplet-pretrained case study instantiates one specific cell of the taxonomy developed in the review: the metric-learning / triplet family, applied under the frozen-encoder fine-tuning protocol described in Section 4.4. This section revisits the experimental results of Section 5 in light of that taxonomy, examining what the results confirm about the triplet-loss family specifically, what they leave open about the other five families reviewed in Section 3, and where the comparative claims of Table 1 are supported, qualified, or left untested by a single-dataset, single-loss case study.



(a) MSE

(b) MAE

Figure 1. Test-set MSE (a) and MAE (b) of RUL predictions from the triplet-pretrained model versus the non-pretrained baseline, as a function of the number of labeled fine-tuning shots. The x-axis is not linearly scaled: shot counts from 2 to 10 are spaced at unit intervals, while 20, 50, and Full are plotted with larger, compressed spacing between them to keep the low-shot region, where the performance gap is largest, readable.

6.1. What the Results Confirm About the Metric-Learning Family

The results in Section 5.3 are consistent with the core assumption attributed to the metric-learning family in Table 1: that a monotonic, ordinal degradation progression, combined with a temporal-proximity sampling rule (Section 4.3), is sufficient to organize an embedding space that transfers well to few-shot RUL regression on C-MAPSS. The consistent MSE and MAE improvement over the non-pretrained baseline across the 2–50 shot range (Figs. 1a and 1b) directly supports the claim, made in Section 3.4, that the triplet loss single-negative-per-anchor design is sufficient to induce useful structure without the larger negative batches required by the contrastive instance-discrimination family (Section 3.3). This is a comparatively favorable result given the triplet loss's lower design complexity relative to the contrastive and multi-view families, as summarized in Table 1.

The narrowing of this gap as the number of fine-tuning shots increases toward the full-data setting is also consistent with the review's framing of pre-training as a solution to a *label-scarcity* problem specifically, rather than a universal architectural improvement: once $|\mathcal{D}_{\text{lab}}|$ is no longer the binding constraint, the randomly initialized baseline is able to learn an equally effective mapping directly from labeled data, and the advantage conferred by the pre-trained embedding space is, by construction, only realized when labels are scarce. This matches the general pattern reported across the SSL-for-RUL literature reviewed in Section 3 (Vermelin et al., 2022; Krokotsch et al., 2022; B. Liu et al., 2025), though notably most of those works report results only at a small number of discrete shot levels; the more fine-grained sweep from 2 to 50 shots reported here (Section 5) makes the diminishing-returns pattern visible as a continuous trend rather than a single before/after comparison.

6.2. What the Case Study Does Not Test

Because the case study isolates a single loss family, a single architecture, a single dataset, and a single fine-tuning protocol (frozen-encoder), several comparative claims in Section 3.8 remain untested by our experiments and are instead inherited from the broader literature reviewed in Section 3.

First, the case study does not directly compare the triplet loss against the contrastive instance-discrimination family (Section 3.3) or the reconstruction family (Section 3.5) under identical conditions. Kong et al. (Kong et al., 2023) and Guo et al. (Guo et al., 2022) report favorable few-shot RUL results using contrastive and masked-reconstruction objectives, respectively, but on different architectures, sampling protocols, and (in the case of Guo et al.) a different sensor domain (machine-tool data rather than turbofan simulation). Whether the metric-learning family's advantage in our experiments would persist, widen, or narrow relative to these alternatives under a matched architecture and protocol is an open question that this case study, by design, cannot answer; it would require the kind of controlled cross-family comparison identified as a gap in Section 3.8.

Second, the frozen-encoder fine-tuning choice adopted in Section 4.4 was selected specifically to isolate representation quality, but several methods reviewed in Section 3, including DCSSL (Shen et al., 2026) and the multisensor contrast framework (B. Liu et al., 2025), instead allow the encoder to

continue adapting during fine-tuning. It is possible that allowing joint adaptation would narrow the full-data convergence observed in Section 5.3 even further, or alternatively that it would preserve a small residual advantage for the pre-trained model by allowing it to fine-tune from a better initialization rather than a random one; our frozen-encoder design cannot distinguish between these possibilities.

Third, the single-channel, single-dataset nature of the case study means it provides no evidence on the multi-view and cross-sensor contrastive family (Section 3.6), whose central assumption, namely the availability of multiple correlated sensor channels, does not apply to the univariate or moderately multivariate treatment adopted here. Likewise, the case study provides no evidence on the meta-learning family (Section 3.7), which departs from the pretrain-then-fine-tune paradigm entirely and requires a distribution of labeled source tasks that falls outside the unlabeled-pretraining premise of the present case study.

6.3. Implications for Loss Function Selection in Practice

Read together, the review and the case study suggest a practical takeaway that is more nuanced than "self-supervised pre-training helps." The magnitude of benefit, and indeed the appropriate choice of loss family, appears to depend on where a given deployment sits along at least three axes surfaced in Section 3.8: the number of available sensor channels (favoring multi-view contrastive losses when channels are plentiful and correlated, or the single-view families otherwise), the plausibility of a monotonic degradation assumption (favoring the triplet and temporal-order families when it holds, and the augmentation-based contrastive or reconstruction families when it does not), and the availability of some labeled task diversity beyond the target asset (favoring meta-learning approaches when present). Our case study occupies a specific and comparatively favorable point in this space, namely a setting where the monotonic assumption is well justified by the C-MAPSS simulation's construction, which likely contributes to the strength of the reported low-shot improvements and should temper expectations of similarly large gains in settings where degradation is less cleanly monotonic, such as intermittent or self-healing degradation processes.

6.4. Open Challenges Highlighted by the Case Study

The case study also surfaces two open challenges beyond those already identified in Section 3.8. First, the convergence of the pre-trained and non-pretrained models under full-data fine-tuning (Section 5.3) raises the question of whether any of the reviewed loss families offer a benefit that persists even when labels are abundant, for instance in the form of improved robustness to distribution shift or reduced variance across random initializations; the present protocol, which reports only point estimates of MSE and MAE, does not address this. Second, our implementation tunes the sampling hyperparameters w_{pos}, $g_{\min}$, and $g_{\max}$ (Section 4.3) specifically for C-MAPSS; the review in Section 3 notes that several other loss families carry analogous, dataset-specific tuning requirements (augmentation strength in contrastive losses, masking ratio in reconstruction losses, task-construction heuristics in meta-learning), and a systematic study of how sensitive each family is to such tuning, and how transferable good settings are across datasets, remains absent from the literature surveyed here.

These considerations motivate the concluding remarks of Section 7, which summarize the paper's contributions and outline the controlled cross-family comparison that would be needed to resolve the open questions raised above.

7. CONCLUSION

This paper addressed the lack of a systematic, loss-function-centric comparison in the self-supervised learning literature for remaining useful life (RUL) prediction. We formalized the shared few-shot RUL problem (Section 2), reviewed six families of self-supervised loss functions used in prognostics, namely temporal-order, contrastive instance-discrimination, metric-learning, reconstruction, multi-view, and meta-learning objectives, summarizing each family's assumptions, requirements, and limitations (Section 3), and instantiated the metric-learning family through a triplet-margin-loss method, validated under few-shot fine-tuning on C-MAPSS (Sections 4 and 5).

The proposed method consistently outperformed a non-pretrained baseline across 2 to 50 labeled shots, with the gap narrowing as labeled data became abundant, confirming that self-supervised pre-training is most valuable precisely in the label-scarce regime that motivates this review. As discussed in Section 6, this supports our taxonomy's claim that a monotonic degradation assumption paired with temporal-proximity sampling yields a transferable embedding space at low design complexity, though the result was validated for a single loss family on a single dataset.

The comparative claims in Table 1 therefore still rest largely on cross-paper comparisons rather than a single controlled study. Future work should conduct a controlled cross-family comparison, holding architecture, dataset, and fine-tuning protocol fixed while varying only the loss function, and extend evaluation to multi-sensor and non-monotonic degradation settings where the multi-view and meta-learning families become directly relevant.

REFERENCES

Chang, L., & Lin, Y.-H. (2025). Few-shot remaining useful life prediction based on bayesian meta-learning with predictive uncertainty calibration. *Engineering Applications of Artificial Intelligence*, *142*, 109980.

Dong, J., Wu, H., Zhang, H., Zhang, L., Wang, J., & Long, M. (2023). Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, *36*, 29996–30025.

Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., & Guan, C. (2021). Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*.

Faysal, A., Ngui, W., Lim, M., & Leong, M. (2023). Ensemble augmentation for deep neural networks using 1-d time series vibration data. *Journal of vibration engineering & technologies*, *11*(5), 1987–2011.

Franceschi, J.-Y., Dieuleveut, A., & Jaggi, M. (2019). Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, *32*.

Guo, H., Zhu, H., Wang, J., Vadakkepat, P., Ho, W. K., & Lee, T. H. (2022). Masked self-supervision for remaining useful lifetime prediction in machine tools. In *2022 ieee 20th international conference on industrial informatics (indin)* (pp. 353–358).

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 16000–16009).

Hou, G., Xu, S., Zhou, N., Yang, L., & Fu, Q. (2020). Remaining useful life estimation using deep convolutional generative adversarial networks based on an autoencoder scheme. *Computational Intelligence and Neuroscience*, *2020*(1), 9601389.

Kong, Z., Jin, X., Xu, Z., & Chen, Z. (2023). A contrastive learning framework enhanced by unlabeled samples for remaining useful life prediction. *Reliability Engineering & System Safety*, *234*, 109163.

Krokotsch, T., Knaak, M., & Gühmann, C. (2022). Improving semi-supervised learning for remaining useful lifetime estimation through self-supervision. *International Journal of Prognostics and Health Management*, *13*(1).

Lai, J.-J., Pan, S.-J., Ong, C.-W., & Chen, C.-L. (2022). Weibull reliability regression model for prediction of bearing remaining useful life. In *Computer aided chemical engineering* (Vol. 49, pp. 829–834). Elsevier.

Li, Y., Bai, X., Liu, C., Chu, J., & Tan, J. (2025). A self-supervised assisted label-efficient method for online remaining useful life prediction. *Measurement*, *242*, 115902.

Liu, B., Xu, Z., Kang, Y., Cao, Y., & Zhao, Y. (2025). Multisensor contrast neural network for remaining useful life prediction of rolling bearings under scarce labeled data. *Frontiers of Information Technology & Electronic Engineering*, *26*(7), 1180–1193.

Liu, J., & Chen, S. (2024). Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 38, pp. 13918–13926).

Ma, L., & Zhang, Z. (2026). A meta multi-scale spatial attention network for interpretable few-shot remaining useful life prediction. *Engineering Research Express*, *8*(7), 075206.

Mo, R., Zhou, H., Yin, H., & Si, X. (2025). A survey on few-shot learning for remaining useful life prediction. *Reliability Engineering & System Safety*, *257*, 110850.

Mo, Y., Li, L., Huang, B., & Li, X. (2023). Few-shot rul estimation based on model-agnostic meta-learning. *Journal of Intelligent Manufacturing*, *34*(5), 2359–2372.

Morales-Espejel, G., & Gabelli, A. (2020). A model for rolling bearing life with surface and subsurface survival: Surface thermal effects. *Wear*, *460*, 203446.

Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012). Pronostia: An experimental platform for bearings accelerated degradation tests. In *Ieee international conference on prognostics and health management, phm'12.* (pp. 1–8).

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2022). A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Paris, P., & Erdogan, F. (1963). A critical analysis of crack propagation laws. *Journal of Basic engineering*, *85*(4), 528–533.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (pp. 1–9).

Shang, X., Shang, J., Li, M., Qiu, H., Gao, L., & Xu, D. (2025). A cross-domain few-shot remaining useful life estimation framework based on model-agnostic meta-learning with task embeddings. *Computers in Industry*, *173*, 104396.

Shen, Z., Yang, C., Cheng, L., Kong, X., Liu, Z., & Su, K. (2026). A novel dual-dimensional contrastive self-supervised learning-based framework for rolling bearing remaining useful life prediction. *Scientific Reports*.

Shi, Z., & Chehade, A. (2021). A dual-lstm framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety*, *205*, 107257.

Song, L., Zheng, C., Lu, X., Liu, J., & Wang, H. (2026). A multi-task prediction network with dynamic convolution and self-supervised contrastive learning in mechanical remaining useful life prediction. *Journal of Vibration and Control*, 10775463261450874.

Tonekaboni, S., Eytan, D., & Goldenberg, A. (2021). Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*.

Vermelin, W. S., Lövberg, A., & Kyprianidis, K. (2022). Self-supervised learning for efficient remaining useful life prediction. In *Annual conference of the phm society* (Vol. 14).

Wang, B., Lei, Y., Li, N., & Li, N. (2018). A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on reliability*, *69*(1), 401–412.

Wang, L., Cao, H., Xu, H., & Liu, H. (2022). A gated graph convolutional network with multi-sensor signals for remaining useful life prediction. *Knowledge-Based Systems*, *252*, 109340.

Wu, C., He, J., Wang, L., Ma, C., & Liu, Y. (2025). A dual-path architecture based on time series decomposition and degradation correction for remaining useful life prediction of aero-engine. *Computers & Industrial Engineering*, *203*, 110964.

Yang, C., Cheng, L., Shen, Z., Liu, Z., Su, K., Kong, X., ... Wang, Z. (2026). Iaa-crl: A universal information-adaptive augmentation based contrastive representation learning framework for remaining useful life prediction of rolling bearings. *Mechanical Systems and Signal Processing*, *254*, 114357.

Yang, J., & Wang, X. (2024). Meta-learning with deep flow kernel network for few shot cross-domain remaining useful life prediction. *Reliability Engineering & System Safety*, *244*, 109928.

Yang, J., Wang, X., & Luo, Z. (2024). Few-shot remaining useful life prediction based on meta-learning with deep sparse kernel network. *Information Sciences*, *653*, 119795.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., & Xu, B. (2022). Ts2vec: Towards universal representation of time series. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 36, pp. 8980–8987).

Zhang, J., Luo, H., Hu, J., Cheng, C., Wu, S., Yan, P., & Tian, J. (2025). A transferable topology-aware graph pooling network for remaining useful life prediction under cross-domain conditions. *IEEE Transactions on Reliability*.

Zhang, W., Li, C., Peng, G., Chen, Y., & Zhang, Z. (2018). A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical systems and signal processing*, *100*, 439–453.

Zhao, Y., & Wang, Y. (2021). Remaining useful life prediction for multi-sensor systems using a novel end-to-end deep-learning method. *Measurement*, *182*, 109685.

Zhou, L., & Wang, H. (2024). Mst-gat: A multi-perspective spatial-temporal graph attention network for multi-sensor equipment remaining useful life prediction. *Information Fusion*, *110*, 102462.

Zhu, Z., Lv, Z., & Xie, T. (2025). Trend contrast features-based bearing remaining useful life prediction method. *Control Engineering Practice*, *162*, 106358.

BIOGRAPHIES

Mohammad Badfar is a Ph.D. candidate in Industrial Engineering at Wayne State University. He received his MSc (2017) degree from Sharif University of Technology and his BSc (2014) from the University of Tehran. In 2021, he joined Wayne State University as a researcher, focussing on monitoring, prognostics, and maintenance planning of energy assets, with a focus on adaptability and generalizability. His research includes anomaly detection in connected vehicle systems, battery state-of-charge estimation, and attention-based maintenance planning. Prior to academia, he worked in industry as a business analyst and systems engineer.

Murat Yildirim is an assistant professor in the Department of Industrial and Systems Engineering, and the Director of Cyber Physical Systems Laboratory at Wayne State University. He obtained a PhD degree in industrial engineering, MSc degree in operations research, and BSc degrees in electrical and industrial engineering from Georgia Institute of Technology. Dr. Yildirim's research interest lies in advancing the integration of mathematical programming and data analytics in various application domains. Specifically, he focuses on the modelling and the computational challenges arising from the integration of real-time inferences generated by advanced data analytics and simulation into large-scale mathematical programming models used for optimising and controlling networked systems. His research has been supported through multiple projects funded by NSF, DoE, MTRAC and Ford.

Ratna Babu Chinnam is a Professor and Chair of the Industrial & Systems Engineering Department at Wayne State University, specializing in AI, Big Data, and Business Analytics. He has authored over 150 technical publications and serves as an Associate Editor for multiple journals. His research spans operations management, supply chains, healthcare, and smart engineering systems, with funding from NSF, DoD, DoE, and industry collaborations with Ford, Intel, and General Dynamics. He has mentored numerous PhD students, many in academia, and received awards for outstanding mentorship and teaching. He holds a PhD in Industrial Engineering from Texas Tech University.