Intelligent fatigue damage tracking and prognostics of composite structures utilizing raw images via interpretable deep learning

Komninos, P.; Verraest, A.E.C.; Eleftheroglou, N.; Zarouchas, D.

DOI
10.1016/j.compositesb.2024.111863

Publication date
2024

Document Version
Final published version

Published in
Composites Part B: Engineering

Composites Part B 287 (2024) 111863

Contents lists available at ScienceDirect

Composites Part B

journal homepage: www.elsevier.com/locate/composites

Intelligent fatigue damage tracking and prognostics of composite structures utilizing raw images via interpretable deep learning

P. Komninos [a,*], A.E.C. Verraest [a], N. Eleftheroglou [a,b], D. Zarouchas [a]

[a] Center of Excellence in Artificial Intelligence for Structures, Prognostics & Health Management, Aerospace Engineering Faculty, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands

[b] Intelligent Sustainable Prognostics Group, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands

ARTICLE INFO

Dataset link: https://data.mendeley.com/datasets/ky3gb8rk9h/1

Keywords:
Interpretable deep learning
Attention mechanism
Spatiotemporal decomposition
Remaining useful life
Crack propagation
Composite structure

ABSTRACT

In recent years, prognostics gained attention in various industries by optimizing maintenance, boosting operational efficiency, and preventing costly downtime. Central to prognostics is the Remaining Useful Life (RUL), representing the critical time before system failure. Deep learning advancements facilitate RUL forecasting by extracting features from diverse data formats such as time series, images, or sequences thereof, in one, two, or three dimensions, respectively. Yet, predicting RUL from image sequences often relies heavily on resource-intensive techniques like digital image correlation, complicating data acquisition. To address challenges with high-dimensional data and unreliable models, this study introduces ISTRUST, an innovative Transformer-based architecture. ISTRUST (Interpretable Spatiotemporal TRansformer for Understanding STructures) tackles the dual challenges posed by high-dimensional data and the black-box nature of existing models. Leveraging Transformers' attention mechanism, ISTRUST breaks down the spatiotemporal domain, effectively realizing interpretable RUL predictions under uncertainty using only sparse raw image sequences as input. Evaluated on fatigue-loaded composite samples showcasing crack propagation, ISTRUST interprets the relation between cracks and RUL via the attention mechanism. The results substantiate its capacity to interpret and clarify instances in which predictions may exhibit variability in accuracy. Through the attention mechanism, a strong correlation between the model's spatiotemporal focus and the RUL predictions is established, making it, to the best of our knowledge, the first model to provide interpretable stochastic RUL predictions directly from sequential images of this nature.

1. Introduction

In today's rapidly evolving technological landscape, the reliability and performance of complex systems are of paramount importance. From aerospace and automotive industries to manufacturing and healthcare sectors, the seamless operation of critical systems is not only a matter of economic significance but also a concern for safety, sustainability, and efficiency. The discipline of Prognostics and Health Management (PHM) has emerged as a key enabler in ensuring the continuous and optimal functioning of these systems by providing early insights into their health status and predicting potential failures [1].

The term "PHM" encompasses a set of multidisciplinary approaches and tools designed to monitor, assess, and manage the health of systems and their components in real-time [2]. By integrating data-driven analytics, advanced sensors, Machine Learning (ML), and domain knowledge, PHM offers a proactive means of identifying anomalies [3–5], predicting impending failures [6–8], and prescribing timely maintenance or remedial actions [9–11]. As such, PHM has the potential to revolutionize the way industries approach maintenance, improving asset utilization, reducing downtime, and minimizing life cycle costs.

One of the central pillars of PHM that has garnered increasing attention in recent years is the estimation of Remaining Useful Life (RUL). While PHM encompasses a broad spectrum of techniques for monitoring and assessing system health, RUL estimation stands out as a critical component in the quest for enhanced system reliability and performance optimization. The concept of RUL can be succinctly defined as the remaining time a system or component can be expected to operate within acceptable performance and reliability limits before experiencing failure or degradation beyond tolerable levels. Under the expansive umbrella of PHM, the estimation of RUL has seen significant advancements, particularly in the context of data-driven

* Corresponding author.
E-mail addresses: P.Komninos@tudelft.nl (P. Komninos), A.E.C.Verraest@student.tudelft.nl (A.E.C. Verraest), n.eleftheroglou@tudelft.nl (N. Eleftheroglou), D.Zarouchas@tudelft.nl (D. Zarouchas).

https://doi.org/10.1016/j.compositesb.2024.111863
Received 15 April 2024; Received in revised form 15 September 2024; Accepted 24 September 2024
Available online 1 October 2024

approaches. While it has been widely applied to univariate or multivariate time-series data [12,13], such as vibration-based signals, and image data [14,15], it is important to acknowledge that its application to sequential images, which represents high-dimensional time-series data, is still in its nascent stages.

In the realm of univariate and multivariate time series data, RUL estimation has demonstrated remarkable success across various industries. This approach typically involves analyzing historical sensor measurements and operational data to predict when a system or component is likely to reach a predefined End-of-Life (EOL) threshold. Numerous algorithms and models, ranging from classical statistical [10,16,17] and numerical [12,13] methods to sophisticated ML techniques [6,18–22], have been developed to handle this type of data. These approaches have proven invaluable in scenarios where data is collected over time, making them suitable for applications like structural health monitoring, predictive maintenance, and failure prognosis. In the context of crack propagation, this type of data often relates to strains extracted by measuring the structure's crack growth. Existing works considered post-processed strain data as input to predict RUL end-to-end utilizing different deep learning models, such as recurrent neural networks (RNN) and convolutional neural networks (CNN) [23,24]. Despite the simplification of the process of predicting Remaining Useful Life (RUL) through reduced memory requirements and the construction of simple deep learning models, these methods may lead to a loss of detailed information. This is primarily due to the complex patterns inherent in raw images not being fully captured, which necessitates significant computational effort to extract the requisite information, as observed in digital image correlation (DIC). Furthermore, the dependency of strain data on measurement techniques can introduce errors, rendering it less reliable and comprehensive compared to the direct analysis of high-resolution images. This underscores the need to consider the direct utilization of raw image data.

Similarly, the extension of RUL estimation to image data, where predictive models are trained to make RUL predictions based on a single image or snapshot, has shown promise in diverse fields. Applications range from predicting the remaining lifespan of critical components in manufacturing machinery [14,15] to assessing the structural integrity of infrastructure components such as concrete structures [25]. With the emerging field of deep learning, CNN, a specific type of Neural Network (NN), has played a pivotal role in enabling RUL predictions from images, revolutionizing the way assets are monitored and managed.

However, when dealing with video data, such as sequences of images, the landscape becomes more intricate. For RUL estimation, these data mainly consist of video frames, reflecting the damage of the examined structure and how it progresses. While the potential of such data in RUL estimation is evident, the complexity and dimensionality of this information pose unique challenges. Extracting meaningful features, handling spatiotemporal dependencies, and building models capable of predicting RUL accurately from frames remain active areas of research. Consequently, NNs, with their capacity to automatically learn hierarchical representations from raw data, have shown remarkable capabilities in handling the challenges of high-dimensional data. State-of-the-art approaches include a combination of RNN, such as Long Short-Term Memory (LSTM), with CNN to effectively capture the temporal and spatial features, respectively [26,27].

Very recently, the LSTM-CNN approach started being replaced by transformers [28], an advanced technique initially proposed in Natural Language Processing (NLP) that has now been applied in engineering applications as well. In addition to the success of transformers in NLP, advancements have enabled their application in computer vision. [29] proposed vision transformers that split images into patches and forward them as a sequence of linear embeddings to a transformer-encoder. Subsequent developments incorporated the ability to analyze a sequence of images by separating the attention mechanism into a spatial and temporal domain [30–32]. However, these studies either encountered challenges in fully separating the temporal and spatial domains throughout their architecture [30,31] or struggled to reduce the network's depth [32].

The attention mechanism, the main building block of transformers, has given remarkable outcomes in a variety of engineering studies. Concerning RUL [33–38], however, all of these works are considered as lower-dimensional data that are easier to handle. Nevertheless, transformers are gaining increasing popularity in volumetric data as well. From video recognition [39] and object detection [40], to semantic segmentation [41] and damage detection [42], they enable the development of models which have achieved unprecedented levels of performance due to their capacity for capturing important information of the complex input data. More importantly, they have the ability to interpret the results and visualize how the input–output pairs are correlated via the attention weights [43], which is a huge step towards unfolding the black-box barriers of deep learning and being advantageous over the typical LSTM-CNN techniques.

In spite of the substantial progress achieved by transformers in various engineering applications, their utilization in RUL prediction with raw sequential image data as the input remains largely unexplored. This is primarily due to the inherent limitations of raw images, which do not provide adequate or accurate information for reliable predictions. Mitigating this constraint requires the application of extensive image processing techniques, notably the incorporation of DIC to capture pertinent features [44,45]. Additionally, surrogate modeling is employed to augment interpretability, however, it is noteworthy that it may inadvertently oversimplify the complex analysis of structures [46]. These approaches can be correspondingly time and computationally inefficient, and there is no evident connection between the sequence of images and the predictions of RUL. This makes the idea of developing a single model that can handle everything from raw sequential image data to RUL prediction end-to-end by harnessing the interpretability inherent to the attention mechanism. Additionally, the capability to predict RUL directly from raw images enhances the framework's applicability in real-world scenarios where the asset lacks sufficient computational resources for an intricate feature extraction process. Simultaneously, this approach unlocks prognostic possibilities for any system outfitted with a camera.

Although limited work exists on tracking crack damage via image-based techniques, deep learning has been considered on other brittle structures. The authors in [47] proposed a CNN model for detecting crack damage on images of earthquake-affected urban scenes accompanied by crack annotations. Another work considered deep learning for pixel-level crack segmentation on masonry surfaces [48]. The authors developed a CNN model for crack detection on patch- and pixel-level images. By training with different network architectures and utilizing transfer learning they succeeded in accurate crack classification and segmentation. A limitation of these implementations is inherent in the selected deep learning architecture, which relies on CNNs. This reliance may result in a potential loss of precision in crack detection and segmentation, particularly in discontinuous areas of the structure, as CNNs are highly dependent on neighboring pixel values. Furthermore, the aforementioned works, along with others in the literature [49–51], do not address the task of end-to-end crack damage tracking and remaining useful life (RUL) prediction using sequential image data as input. This highlights the novelty of our approach.

While Physics-informed machine learning techniques are available [52,53], they tend to be highly task-specific. The objective of this research is to develop an ML model that achieves both generalizability across various applications and interpretability. Consequently, this study proposes a novel, interpretable deep learning model – namely Interpretable Spatiotemporal TRansformer for Understanding STructures (ISTRUST) – for RUL prediction from raw sequential image data based on the attention mechanism and the transformer-encoder architecture. As the predictions are now wholly contingent upon raw data inputs, the ISTRUST model possesses the capability to discern scenarios in which predictions may be suboptimal and when they are

poised to demonstrate efficacy. Thus, it is reasonable to anticipate that the model's performance may not surpass that of DIC or surrogate modeling. Instead, its principal objective is to offer insights and comprehension regarding the quality of predictions, thereby achieving an unparalleled level of reliability and interpretation within the context of black-box models. The interpretability is achieved by decomposing the spatiotemporal continuum into spatial and temporal dimensions. Therefore, the spatial attention weights show the correlation between the cracks of each image separately and the predicted RUL, while the temporal attention weights indicate the contribution of each frame to the corresponding RUL. To the best of the authors' knowledge, this work is the first to provide interpretable RUL predictions directly from sequences of raw images and the model's performance is demonstrated with an experimental dataset acquired by composite samples that are under fatigue loads with visible cracks that propagate with time [54]. Finally, since RUL is a random variable, the proposed ISTRUST model considers the uncertainty of the predictions by integrating the Monte Carlo (MC) dropout technique [55] into the overall process.

In summary, the contribution and novelty of the current study to the corresponding research areas is highlighted as follows:

- The model, which consists of a combination of CNN and self-attention layers, performs RUL prediction under uncertainty given unprocessed raw sequential images as input of composite samples that is under fatigue loads.
- This approach contradicts typical models by unveiling the black-box barrier through its integrated interpretation. The model's interpretability is further improved by the effective decomposition of the spatiotemporal domain.
- Interpretability has been achieved by utilizing only one transformer-encoder layer for each domain, deviating from the conventional approach of employing 5–10 layers in vision transformers.
- While it is acknowledged that the predictive performance of the model may not be optimal in certain specimens, it is imperative to note that our method offers a rational and coherent explanation for the underlying phenomena contributing to this observation, paving the way for visionary and innovative concepts for further improving its efficacy.
- We incorporate supervised contrastive learning to compel the model to filter out irrelevant information, such as the small variations in the characteristics between each specimen, and the projected data augmentation technique [56–58] that helps in the training process. Moreover, we validate the model's performance to effectively filter out irrelevant information via a Uniform Manifold Approximation and Projection (UMAP) representation [59]. Utilizing contrastive learning we managed to train a vision transformer-inspired architecture on limited data.

The paper is consequently organized as follows: Section 2 describes the methodology from vanilla NN to transformers, self-attention, and MC dropout for uncertainty quantification, along with the data acquisition strategy, experimental setup, and damage propagation phenomena in composite materials. In Section 4, the architecture of the model is presented, visualized, and extensively explained, accompanied by several training strategies to tackle the limited size of the dataset. Finally, the results coming after the model training are shown in Section 5, followed by the conclusions in Section 6.

2. Theory

2.1. Transformers & multi-head attention

NN is a common tool used in machine learning to learn patterns from data. An NN consists of an input layer, optional hidden layers, and an output layer. A Fully Connected (FC) layer consists of several neurons, which aggregate and process the output of the previous layers linearly via matrix multiplications. An NN with at least one hidden FC layer and a non-linear activation function for that hidden layer is called a multilayer perceptron (MLP). Prior to applying a forward propagation, the weights and biases are initialized; typically the weights are sampled from a normal or uniform distribution and the biases are zeroed. Using the backward propagation algorithm [60], where the error is back propagated through the NN layers, the gradients of the weights and biases w.r.t. the error can be calculated. Using these gradients and an optimizer like Stochastic Gradient Descent (SGD), the weights and biases can be updated accordingly.

MLPs are a very inefficient way of handling images as they require flattening the image to a one-dimensional vector, resulting in an input layer with a large size, and consequently a large amount of weights. Furthermore, MLPs lack the ability to learn a specific feature irrespective of its position within an image. Consequently, MLPs must repetitively learn the same feature for each location individually, thus lacking translational invariance. For an image $\mathbf{x} \in \mathbb{R}^{K\times H\times W}$, where K is the number of features (e.g. 3 in the case of an RGB image), H is the height of the image in pixels and W is the width of the image in pixels, CNN solve the problem of a large number of weights and the translational invariance by utilizing a different aggregation technique, i.e. a filter or kernel called 2D convolution [61,62].

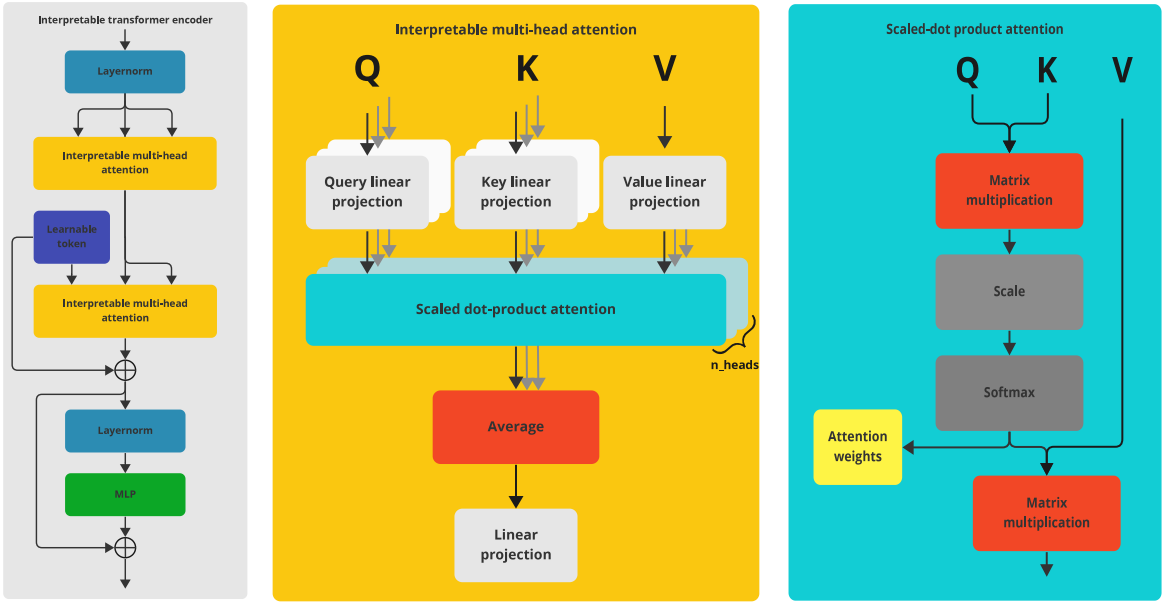
To further improve the models' performances, the transformer architecture was proposed by Vaswani et al. [28]. By processing sequential data using self-attention [63], they eliminated the need for RNNs or CNNs. Transformers consist of an encoder, processing the input data, and a decoder, generating the output data. The encoder and decoder consist of sequential interchanged self-attention and MLP layers and are connected through the encoder–decoder attention mechanism. Following the complexity of the original transformer, Devlin et al. [64] successfully advocated for utilizing only the transformer-encoder, resulting in reduced complexity and enabling improved interpretability. In this regard, we consider the proposed architecture as a fundamental stepstone for constructing our ISTRUST model, capable of fulfilling our requirements.

Self-attention acts as the fundamental component of transformers, in which queries (**Q**), keys (**K**), and values (**V**) are computed based on the input data. The queries, keys, and values are obtained by means of a linear projection from a sequence of one-dimensional input vectors with size d_{model}. Subsequently, they all undergo a scaled dot-product attention (see also Fig. 1(c)):

$$Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = softmax\left(\frac{\mathbf{QK^T}}{\sqrt{d_{model}}}\right)\mathbf{V} \tag{1}$$

where $\sqrt{d_{model}}$ is a scaling factor. Initially, the queries and keys are multiplied by the means of a dot product between each other. After scaling and applying the softmax function, the result is a matrix of attention weights. Finally, these attention weights are a matrix multiplication with the values, effectively collapsing the number of keys and the corresponding associated information, represented by the values.

Nevertheless, since the size of the queries and keys is typically in the order of a hundred, the computation of the attention weights is computationally expensive. Rather than downscaling d_{model}, which would sacrifice information, the queries and keys are split by performing multiple linear projections, resulting in multiple heads, as shown in Fig. 1(b). The resulting projected queries and keys have a size of $d_k = d_{model}/n_{heads}$, where n_{heads} represents the total number of heads. Typically, the values are also divided among the heads. However, this approach would compromise interpretability, as it would introduce an indeterminate level of relative importance among the heads. For that reason, a variant of multi-head attention called interpretable multi-head attention is implemented [65]. In interpretable multi-head attention, the values all undergo the same linear projection to achieve the projected values. Unlike the queries and keys, the values do not pose a computational bottleneck and can retain their original size of d_{model}. Following the scaled-dot product attention in each head, the



(a) Interpretable transformer-encoder

(b) Interpretable multi-head attention

(c) Scaled dot-product attention

Fig. 1. Transformer-encoder. Fig. 1(a) shows the fundamental part of the ISTRUST model, i.e. the interpretable transformer encoder. Fig. 1(b) depicts the interpretable multi-head attention. Fig. 1(c) shows the scaled-dot product attention.

resulting attended outputs are averaged. Next, the averaged outputs are passed through a linear projection to finally obtain the attended vectors.

As depicted in Fig. 1(a), the interpretable transformer-encoder is distinguished from the typical architecture due to the interpretable multi-head attention mechanism. In our proposed architecture, there are two interpretable multi-head attention blocks. Similarly to the original transformer-encoder, the embedding vectors produced by computing the inputs (in our case, a sequence of images), are initially normalized using layer normalization [66] and then are consequently fed to the first interpretable multi-head attention block. Next, they are passed through the second multi-head attention block alongside a learnable token, which will be further described in Section 2.2. This learnable token has additionally the role of a residual which is added to the embedding vectors produced by the multi-head attention. The final constructed embedding vectors are once again normalized using layer normalization, whereafter they are passed through an MLP. In the MLP block, there is no interaction or information transfer between the embedding vectors. Instead, they are transformed within their own dimension of d_{model}. After the MLP, there is another summation through a residual connection.

2.2. Vision transformers

Following the revolution in NLP, transformers are also being applied in computer vision tasks [29]. Vision transformers, as an alternative to traditional CNNs, have gained popularity due to their improved performance. Passing all pixels of an image into a vision transformer – which is based on the encoder part of the original transformer – to the self-attention mechanism would require computational power that is out of the limits of today's hardware. Therefore, the input image is divided into square patches. Since transformers usually take one-dimensional vectors as an input, the two-dimensional patches are either flattened and linearly projected or are fed into a CNN to obtain one-dimensional embedding vectors of size d_{model} [67]. Since the interpretable multi-head attention as shown in Fig. 1(b) does not change the dimensionality of its input, the information present in the patches cannot be encoded into a lower dimensionality. This implies that the contribution of each patch to the outputs cannot be detected. One solution is to flatten each patch and stack those one-dimensional arrays together into one large vector. However, this can quickly become memory-inefficient, time-consuming, and the patches will lose the information about their pixels' relative positions. To address these issues and to keep the relative importance of each patch, another solution is to use a learnable parameter that contains information about their contribution to the architecture. For that reason, a learnable token is concatenated to the embeddings, also called a CLS token in classification problems [29]. This learnable token is a one-dimensional vector of size d_{model} with learnable parameters that, similar to the weights in an NN, are updated through backpropagation. The learnable token is a crucial component of the vision transformer as it focuses on capturing the essential information within the image patches that is relevant to the desired output.

Unlike CNN, a transformer is inherently not aware of the location of a patch. In this regard, it was proposed in [28] to use a technique called positional encoding. This method involves adding a one-dimensional vector, dependent on the location, to the embedding vector. Various methods exist for calculating these positional encoding vectors, but a common approach involves using sine and cosine functions. Considering the one-dimensional embeddings with their relative position encoded, they are fed into the transformer-encoder together with the learnable token. In the transformer-encoder, they undergo several interchanged self-attention and MLP layers. In the self-attention layer, the learnable token represented as a query searches for particular features, in our case RUL-related features, in the corresponding patches represented by the keys. The values, on the other hand, contain the information present in a patch that is ideally relevant to extract features with the queries using a dot-product.

Following the transformer-encoder, only the attended embedding vector originating from the learnable token is utilized for further processing, while the embedding vectors from the patches are discarded since their relevant information is encoded into the learnable token. Finally, the embedding vector originating from the learnable token is passed through another MLP with, as output dimension, the number of classes for classification problems or the number of outputs for regression problems.

To sum up, each input image is first divided into patches. Subsequently, patch embedding is applied followed by positional encoding. The transformer-encoder encodes the information pertinent to the specific problem into the learnable token using the interpretable multi-head attention. Following the transformer-encoder, the patch embeddings are discarded and the attended learnable token undergoes an MLP to achieve the final prediction.

3. Problem formulation and experimental setup

3.1. Problem formulation

Given input data comprising sequential images of a composite structure subjected to fatigue loads and exhibiting crack growth, the primary task is to predict the RUL of the structure and estimate the associated uncertainty by interpreting the model's focus within the spatial and temporal domains of the input data. In addition to executing a high-dimensional stochastic regression task, efforts are directed towards enhancing the trustworthiness of the model's predictions by visualizing the areas of focus. This approach aims to provide deeper insights into the model's prediction outcomes and improve reliability.

To achieve these objectives, an advanced deep learning technique based on vision transformers is proposed. The model is further refined to highlight and analyze critical regions and time frames that contribute most significantly to the prediction of RUL. The proposed method provides interpretable insights into the degradation process of the composite structure. Additionally, uncertainty quantification is performed using the MC dropout technique [55], which allows for the assessment of Confidence Intervals (CI) around the predicted RUL. This comprehensive approach ensures that the predictions are reliable even when not optimal, facilitating better decision-making in maintenance and safety protocols for composite structures under fatigue loads.

3.2. Experimental setup

This subsection presents the process of data acquisition, which is crucial for understanding the experimental data used in this study. It covers the acquisition of experimental data, including details about the specimens, materials, and the experimental setup. Furthermore, it explores the concepts of crack propagation and the definition of RUL in this context and provides insights into the damage accumulation mechanisms observed in composite structures under fatigue loads. Lastly, it discusses the transformation of the acquired raw images into a dataset suitable for analysis and modeling.

The experimental data used in this paper is provided by a previous experimental study [21]. The material at hand is a unidirectional Prepreg tape Hexply® F6376CHTS(12K)-5-35. The laminate is manufactured using a hand lay-up of $[0/45/90/-45]_{2s}$ and is cured in an autoclave at a temperature of 180 C and pressure of 9 bar for 120 min as recommended by the manufacturer. The laminate is consequently cut to obtain specimens of 400 mm × 45 mm with an average thickness of 2.28 mm. Two examples of specimens can be found in Fig. 2.

Subsequently, the specimens are loaded under fatigue using a 100 kN MTS fatigue controller and a bench fatigue machine, with an average fatigue load of 16.2 kN, a stress ratio of $R = 0.1$, and a frequency of 10 Hz. The experiments are paused every 500 cycles, during which the load is first allowed to go to the minimum load σ_{min}, after which the load is ramped up to σ_{max} over a duration of 1 s. The load then remains stationary at σ_{max} for 2 s. In the middle of this interval, the pictures are taken using two 8-bit "Point Grey" cameras with "XENOPLAN 1.4/23" lenses, placed slightly left and slightly right in front of the specimen. Finally, the load is relieved to σ_{min} over a duration of 1 s, after which the fatigue loading continues. More detailed information about the used materials and the experimental setup can be found in [68]. In the previous study, DIC speckles were utilized to investigate certain characteristics of stiffness, a focus that falls outside the scope of the current research. Our primary objective is to predict RUL from raw images, and therefore, DIC speckles do not contribute to this analysis. Nevertheless, additional noise is introduced on each image, making the tasks of crack tracking and RUL prediction even more challenging.

3.2.1. Crack propagation mechanism and RUL definition

During the fatigue test, several damage accumulation mechanisms can be noticed. Reifsnider and Talug [69] proposed a three-stage process for damage accumulation, describing it as a multistate degradation phenomenon. Fig. 3 provides a visual summary of the damage accumulation process in composite structures, and the damage observed specific to our data is shown in Fig. 2. The process begins with transverse matrix cracking in highly stressed layers, which are shown in blue, followed by the formation of debonding and delaminations at layer interfaces shown in orange in the same figure. In the final stage, fiber failures increase, leading to macroscopic failure, also known as the end of life (EOL). It is important to note that the exact sequence depends on factors such as layup configuration, material properties, manufacturing defects, loading, and environment.

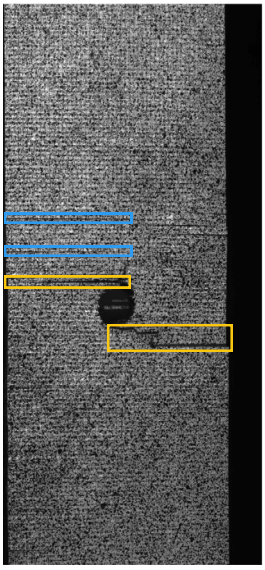
Following the fatigue test, the EOL is marked by the specimen breaking in two, rendering it incapable of bearing any further forces. At each timestep, the period left before reaching the EOL is referred to as the corresponding RUL at this specific timestep. Typically, the RUL is graphically shown as a line with a negative slope, intersecting the x-axis at EOL time.

3.2.2. Dataset setup

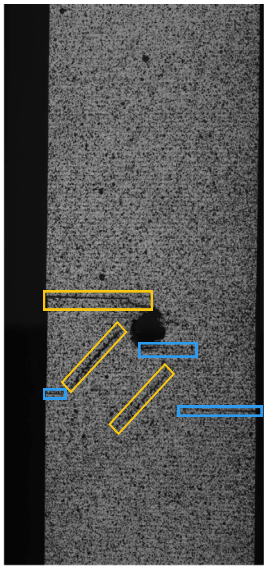
Once the data is acquired, it undergoes a transformation process to prepare it for input into the ISTRUST model. Our data consists of six different specimens, each consisting of two views taken by the two cameras. Per specimen 50 to 150 images are taken, depending on their corresponding total useful life. The raw data is processed as follows:

- For hyperparameter tuning, the dataset is split into a training set, consisting of four specimens, a single validation specimen, and a single testing specimen. The training set is consequently used to train the ISTRUST model, whilst the validation specimen is used to evaluate the performance of the model and adjust the hyperparameters accordingly.
- To evaluate the model with tuned hyperparameters, the validation specimen used for the hyperparameter tuning is discarded to avoid data leakage [71]. All specimens with the exception of the validation specimen are consequently used in cross-validation: the model is trained six times with the same hyperparameters, where each specimen acts as a testing specimen at a certain training iteration. It is important to note that the testing specimens never influence the hyperparameters.
- Rather than using the two different camera views as features, they are split as if they were different specimens. This essentially doubles the length of the dataset, acting naturally as data augmentation and consequently reducing the risk of overfitting.
- The images are sampled from the raw data by utilizing the windowing technique (see Section 4.3 for more details).
- The current RUL of a sample $\mathbf{x} \in \mathbb{R}^{T\times H\times W}$ is defined as the distance between the current time step and the EOL of the specimen which is measured at the moment of the latest picture in a specific array of images.

Based on the abovementioned processes, the training set consists of a range of 500–600 samples of 3 sequential images each, corresponding to an array of images, while the testing set contains a range of 90–130 samples with the same number of sequential images. The corresponding number of samples depends on the specific specimen chosen as the testing set under the concepts of the cross-validation technique.



(a) Sample with multiple horizontal transverse matrix cracks (blue) and delamination as a result of horizontal matrix cracks (orange).



(b) Sample with multiple horizontal transverse matrix cracks (blue) and delamination as a result of horizontal and diagonal matrix cracks (orange).

Fig. 2. Types of cracks; horizontal-only (Fig. 2(a)) and horizontal-diagonal (Fig. 2(b)) in the experimental samples. Two main damage types are observed: transverse matrix cracks and delamination. Transverse matrix cracks – characterized by their slender and sharp nature – are indicated in blue and delamination – characterized by its less slender and more blunt nature – is indicated in orange. Note that delamination is much more visible than the transverse matrix cracks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

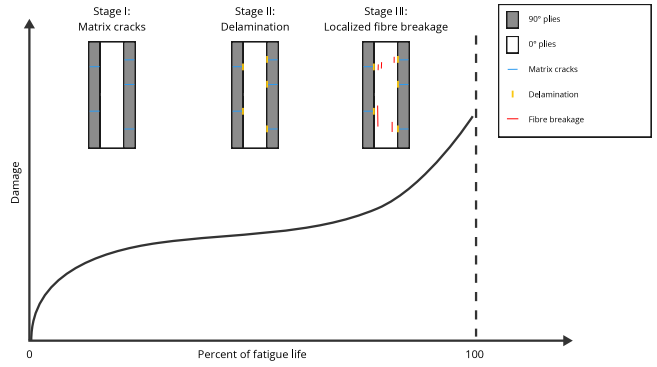


Fig. 3. Three stage damage accumulation process in composite structures [69,70].

4. Model architecture and learning process

The general concept from creating the dataset to predicting RUL with interpretable capabilities is depicted in Fig. 4. Initially, the given data are augmented (Section 4.1) before they are sliced into patches (Section 4.2). Then, the patched images are grouped into small windows of size 3 while 2 images are skipped at every window (Section 4.3). Next, the training process takes effect where our proposed ISTRUST model is trained with two losses; one based on contrastive learning and used for the first part of the learning process, and the typical Mean Squared Error, applied in the second part of the learning process. Finally, the interpretability of the model indicating the correlation between the images and the predictions is illustrated.

In order to provide insights into the connection between input images and RUL predictions, our ISTRUST model incorporates a variation of NN components, such as CNN, self-attention, and MLP techniques. A comprehensive explanation of the model architecture employed for this purpose, the learning process alongside the application of supervised contrastive learning, aimed at filtering out irrelevant information, and its validation using a UMAP representation, can be found in Section 4.4. Moreover, the interpretability of our novel transformer-encoders and the information flow in the model architecture is thoroughly explained using the attention weights in Section 4.5.

4.1. Data augmentation

In order to aid the ISTRUST model in learning useful representations rather than memorizing the data, an augmented training set is created by applying several data augmentation techniques. These techniques are a random resized crop, followed by a random horizontal and vertical flip, and a random rotation of $\pm 5°$ is applied. These augmentations are applied D times on the input images $\mathbf{x}$, resulting in an augmented training set that is D times larger than the original training set. Importantly, the exact same augmentations should be applied to each image in the temporal domain as well. Doing otherwise would confuse the temporal transformer-encoder since the images would no longer be spatially aligned in the temporal domain.

4.2. Patching

The input images $\mathbf{x} \ \epsilon \ \mathbb{R}^{T\times H\times W}$ are split into square patches $\mathbf{x}' \ \epsilon \ \mathbb{R}^{T\times P_H\times P_W\times H_P\times W_P}$, where the number of vertical patches P_H is H/H_P, the number of horizontal patches P_W is W/W_P, and H_P, W_P are the horizontal and vertical pixels of each patch, respectively, as shown in Fig. 7(a). Following the patching, the spatial domain will refer to P_H and P_W rather than H and W. Note that the spatial domain can also be referred to as P in the case that P_H and P_W are flattened into a single dimension, where P is the total number of patches.

4.3. Windowing

The images are sampled from the raw data by shifting a window with a size of s_{window} over the sequence of images. For both performance and overfitting reasons, a skip size s_{skip} can be used, reducing the number of images in a window without reducing the desired window length, as illustrated in Fig. 5. Therefore, a final input size of $\mathbb{R}^{T\times H\times W}$ is obtained, representing an array of images, where $T = \frac{s_{window}-1}{s_{skip}+1} + 1$ is the number of images, H the height, and W the width of the image. In this work, a skip size s_{skip} of 2 and a window size s_{window} of 7 are used.

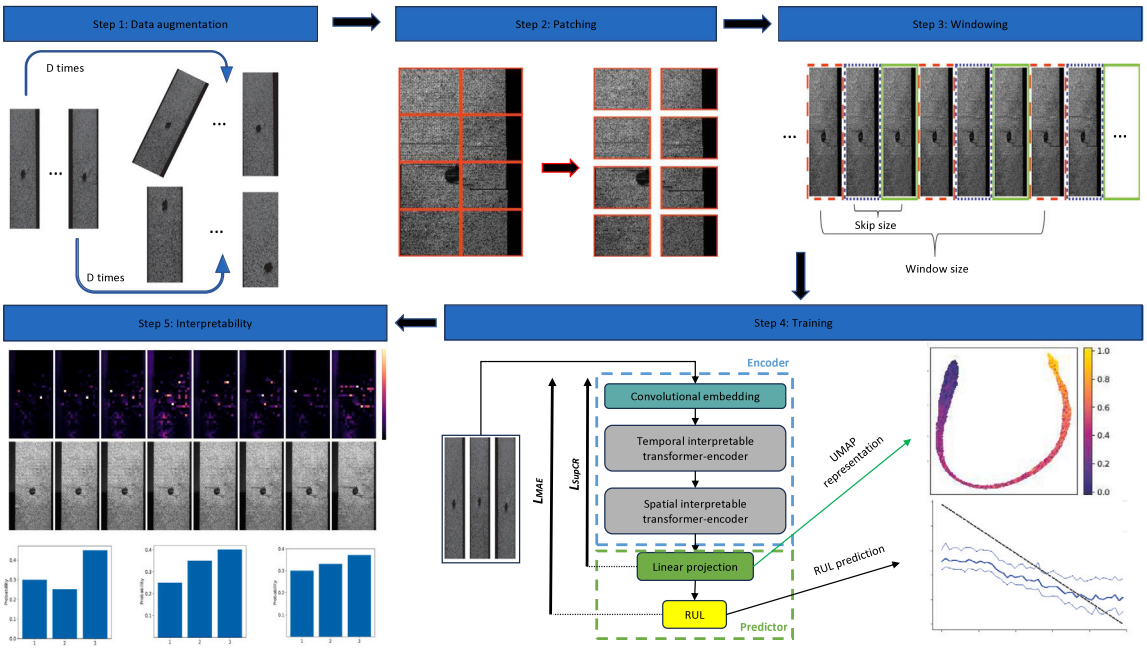


Fig. 4. The general concept concerning the task of predicting RUL under uncertainty from raw sequential images by interpreting the relation between RUL and images in the temporal and spatial domains.

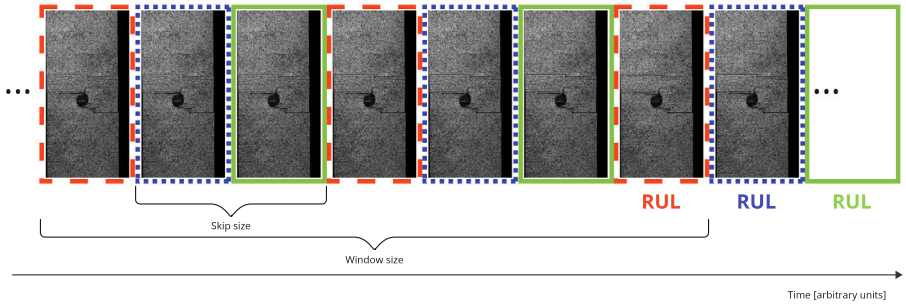


Fig. 5. Sampling images from raw data, for one specimen specifically. Three different samples are shown: the images of the first sample are circled with a red dashed line, the second with a blue dotted line, and the third with a green solid line. The horizontal axis represents the time domain, and consequently decreasing RUL of the specimen. The RUL of the sample is represented by the latest image of the corresponding color and is the distance between the current time step and the EOL of the examined specimen. In this work, a skip size s_{skip} of 2 and a window size s_{window} of 7 are used. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.4. Training setup

Since the task demands high-dimensional data as input representing sequences of images, a spatiotemporal domain prevails. In our custom vision transformer, we introduce a novel approach by decomposing the spatiotemporal domain into distinct domains, namely the temporal and spatial domains. This decomposition enables us to capture and analyze the evolution in both domains separately, leading to improved understanding and interpretability in predicting the RUL from raw images.

The *temporal domain* within our custom vision transformer focuses on the manipulation and rearrangement of information along the T dimension, representing the temporal aspect of the data. This approach empowers us to identify critical temporal patterns, visualize their importance across multiple time steps, and leverage them for accurate RUL predictions. By treating the spatial dimensions H and W as batch dimensions, i.e. dimensions where the individual samples remain isolated without exchanging any information, the temporal domain becomes a dedicated space for uncovering the temporal evolution of the input data.

In conjunction with the temporal domain, our custom vision transformer incorporates the *spatial domain*, which emphasizes operations within the spatial dimensions. By exclusively considering spatial relationships and patterns, we can extract valuable insights into the spatial distribution of image features and their impact on RUL prediction. Through this spatial analysis, we gain a deeper understanding of the material's health status and the location of the damage.

The resulting model architecture used to predict the RUL from the raw images is described in Fig. 6. The temporal and spatial domains represent the first and second interpretable transformer-encoder, respectively. The rationale for prioritizing the decomposition of the temporal domain as a primary consideration, followed by the subsequent analysis of the spatial domain, stems from our inherent emphasis on identifying the most pertinent image and subsequently directing our focus towards specific regions within those images. The individual modules and submodules in the figure will be described in more detail throughout the following subsections. Unless explicitly stated otherwise, operations in one domain exclude operations in the other domain. In other words, when operations are performed in the spatial domain, the temporal domain acts solely as a batch dimension and vice versa.

Provided that the dataset is fairly small – only six specimens, with 50 to 150 images each – a model using multi-head attention like ours

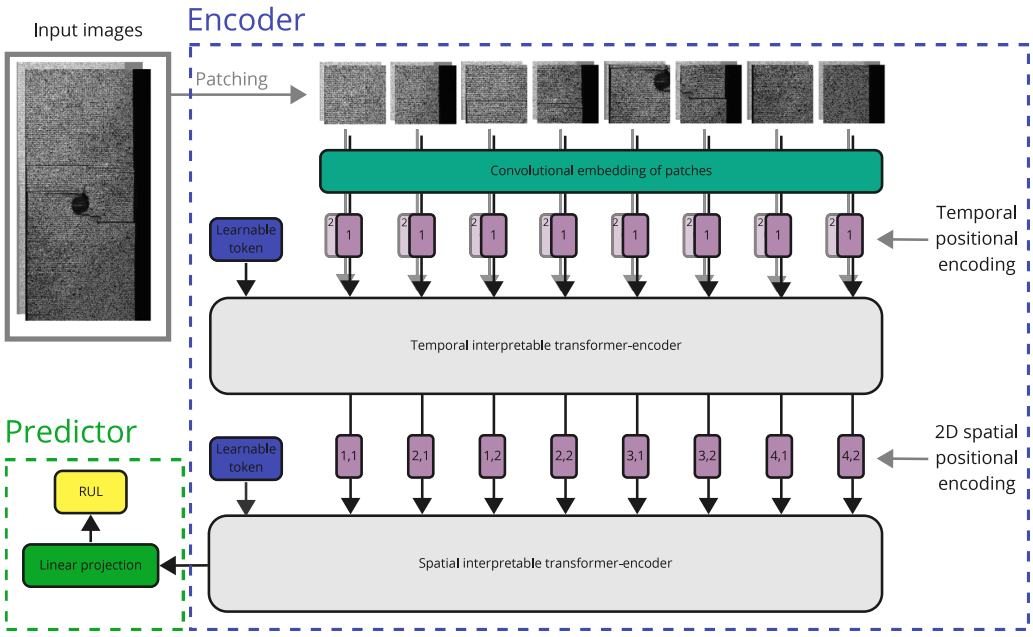


Fig. 6. ISTRUST model's architecture. Our proposed architecture is based on Vision transformer. The images are first divided into patches, followed by an embedding layer and temporal positional encoding. The index i (see Eq. (2)) used for the temporal positional encoding is represented by the numbers in the purple boxes. The patch embedding is followed by the temporal interpretable transformer-encoder, which is shown in more detail in Fig. 1. The temporal interpretable transformer-encoder is followed by 2D spatial positional encoding. The indices i, j (see Eq. (4)) used for the temporal positional encoding is represented by the numbers in the purple boxes. Subsequently, the spatial encoded embeddings are passed through the spatial interpretable transformer-encoder. Finally, the attended learnable token is sent through an MLP to adjust the dimension for the output, i.e. the RUL predictions. Visualizing the temporal and spatial attention weights of the corresponding interpretable transformer-encoder offers valuable insights into the relation between RUL and the speed and size of the structure's cracks, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

has many difficulties learning useful patterns from the data rather than overfitting to the training set, compromising the performance on the testing one. The following techniques are applied to increase the ISTRUST model's performance.

4.4.1. Convolutional embedding

Following the patching layer, the convolutional embedding layer, shown in Fig. 7(b), aims to encode the two-dimensional information of all patches into a one-dimensional embedding vector. It achieves this by performing four stepwise convolutional operations, followed by a batch normalization and GeLU activation function. The convolutional layers essentially reduce the size of the square patch size from shape $H_P \times W_P \times 1$, since the images are grayscale, to shape $1 \times 1 \times d_{model}$. Consequently, these 1×1 dimensions are discarded, with only the features remaining. Subsequent to the convolutional embedding the embedding vectors $\mathbf{z} \in \mathbb{R}^{T \times P_H \times P_W \times d_{model}}$ are obtained, having successfully reduced the 2D patches to a one-dimensional vector. It is important to note that, so far, there has been no information exchange or interaction between the patches in either the temporal or spatial domain. The convolutional embedding operates on each patch independently without considering the spatiotemporal relationships.

4.4.2. Temporal positional encoding

To produce the outputs of the temporal position encoding, i.e. the temporal positional encoded embeddings which should be fed to the temporal transformer-encoder, we use the original positional encoding equation using sine and cosine functions proposed in [28]:

$$TPE(t, 2i) = sin\left(t \cdot e^{\left(2i \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{2}$$

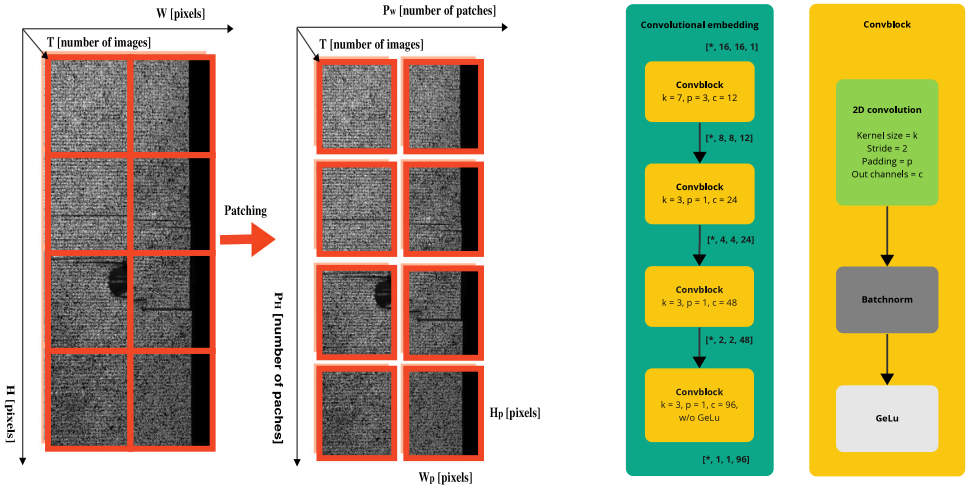
$$TPE(t, 2i+1) = cos\left(t \cdot e^{\left((2i+1) \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{3}$$

where t represents the moment in time and $t \in [0, T)$, $2i$ represents an even index and $2i + 1$ represents an odd index in the embedding vector where $\{2i, 2i+1\} \in [0, d_{model})$. These temporal positional encoding vectors TPE are added to the embeddings $\mathbf{z}$ to obtain the temporal positional encoded embeddings $\mathbf{z}^{TPE}$ $\mathbb{R}^{P_H \times P_W \times T \times d_{model}}$.

4.4.3. Temporal interpretable transformer-encoder

Following the temporal positional encoding, the temporal positional encoding vectors $\mathbf{z}^{TPE}$ are passed through the temporal transformer-encoder where the spatial dimensions P_H and P_W act as batch dimensions. Consequently, information will only be transferred in the temporal domain and not the spatial domain. Traditional vision transformers have multiple self-attention layers being interchanged with MLPs, essentially putting multiple transformer-encoder blocks in series. To achieve interpretability, however, our novel interpretable transformer-encoder shown in Fig. 1(a) consists of two attention mechanisms, namely the multi-head self-attention and the multi-head token-attention, followed by an MLP. The interpretable transformer-encoder block is applied only once in each domain.

After normalizing the temporal positional encoding vectors $\mathbf{z}^{TPE}$ using layer normalization, they undergo the multi-head self-attention layer. However, unlike the conventional vision transformers, the learnable token does not pass through the multi-head self-attention layer. The purpose of the self-attention layer is to facilitate information exchange among the patches. The self-attention layer is followed by the token-attention layer, which is a modified attention layer based on the self-attention in vision transformers [29]. The multi-head token-attention employs the interpretable multi-head attention similar to the self-attention layer (see Section 2.1). Nevertheless, the queries, keys, and values do not come from the same sequence of embeddings. While the queries originate from the learnable token, the keys and values originate from the attended patches that come from the previous self-attention layer, hence the name. The objective of this layer is to encode the information from all embedded patches in the temporal domain into a single vector. As previously mentioned in the description of the interpretable transformer-encoder (see Fig. 1(a)), the attention layers



(a) Patching (b) Convolutional embedding

Fig. 7. Illustration of the patching (Fig. 7(a)) and convolutional embedding process (Fig. 7(b)). In Fig. 7(a), the patched input image is shown. The overlapping images represent the temporal dimension T. For clarity, the image is only separated into a few patches whilst, in practice, the image consists of significantly more patches, where this amount is limited by the hardware on which the model is trained. In Fig. 7(b), the convolutional embedding layer is shown on the left, where the Convblock is illustrated on the right. The shape of the patches is illustrated with square brackets, where $*$ represents an arbitrary number of leading dimensions acting as batch dimensions, which are being left out for clarity. This figure specifically illustrates the convolutional embedding for a 16×16 patch; a smaller patch size would require fewer Convblocks and vice-versa.

are finally followed by layer normalization and an MLP, which in turn, come after another residual connection giving the final temporal encoded embedding vector $\mathbf{z}_{TE} \in \mathbb{R}^{P_H \times P_W \times d_{model}}$, representing the output of the temporal interpretable transformer-encoder.

4.4.4. Spatial positional encoding

Since an image is two-dimensional, utilizing the previous positional encoding technique responsible for the temporal positional encoding (Eq. (2)) would require flattening the spatial domain $[P_H, P_W]$ to $[P]$ and, thus, not differentiating between the horizontal and vertical dimensions. Therefore, an alternative positional encoding technique is performed that incorporates both spatial dimensions, by reserving half of the embedding vector for the positional encoding in the height direction, and the other half for the width direction:

$$SPE(h, 2i) = sin\left(h \cdot e^{\left(2i \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{4}$$

$$SPE(h, 2i+1) = cos\left(h \cdot e^{\left((2i+1) \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{5}$$

$$SPE(w, 2j) = sin\left(w \cdot e^{\left(2j \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{6}$$

$$SPE(w, 2j+1) = cos\left(w \cdot e^{\left((2j+1) \cdot \frac{-ln(10000)}{d_{model}}\right)}\right) \tag{7}$$

where h and w represent the position in the height dimension P_h and width dimension P_w, respectively, with $h \in [0, P_H)$ and $w \in [0, P_W)$, 2i and 2j represent an even index, and 2i + 1, 2j + 1 represent an odd index in the embedding vector where $\{2i, 2i+1\} \in [0, d_{model}/2)$ and $\{2j, 2j+1\} \in [d_{model}/2, d_{model})$. Finally, the temporal positional encoding TPE is added to the temporal encoded embeddings $\mathbf{z}_{TE}$ to obtain the spatial positional encoded embeddings $\mathbf{z}_{TE}^{TPE}$.

4.4.5. Spatial interpretable transformer-encoder

Before the spatial positional encoded embeddings $\mathbf{z}_{TE}^{TPE} \in \mathbb{R}^{P_H \times P_W \times d_{model}}$ can be fed to the spatial interpretable transformer-encoder, the spatial domain has to be flattened to obtain $\mathbf{z}_{TE,f}^{TPE} \in \mathbb{R}^{P, d_{model}}$. Consequently, the spatial interpretable transformer-encoder works entirely similar to the temporal counterpart, except that P_H and P_W – which were previously acting as batch dimensions – are no longer present, and that the T-dimension is replaced by the P-dimension. The corresponding output is consequently the spatial encoded embedding $\mathbf{z}_{SE} \in \mathbb{R}^{d_{model}}$.

4.4.6. Linear projection

The spatial interpretable transformer-encoder is finally followed by a linear projection layer. It encodes the spatial encoded embedding $\mathbf{z}_{SE}$, from $\mathbb{R}^{d_{model}}$ to $\mathbb{R}$ by means of a single linear projection layer followed by a ReLU activation function, giving the predicted RUL.

4.4.7. Weight initialization

On each image, although some diagonal delaminations are observed, the majority is horizontal. This is because most cracks initiate in the 90° ply direction as it is orthogonal to the loading direction. Using the fact that most of these cracks are horizontal, the convolutional embedding layer can be engineered to capture these cracks. Typically, the weights of the kernels, also called filters, in any convolutional layer are sampled from either a uniform or normal distribution, in this case, a normal distribution, giving no preference to horizontal or vertical features, as shown in Fig. 8(a). In our case however, although still sampling from a normal distribution, no variance is allowed in the horizontal direction, thus forcing the kernels to initially filter out vertical features and capture only horizontal features as shown in Fig. 8(b), significantly speeding up the training process and consequently increasing the model's performance. Note that this only happens in the initial stages of the model's training and the kernels are not constrained, hence horizontal variance can still become detectable throughout the learning process.

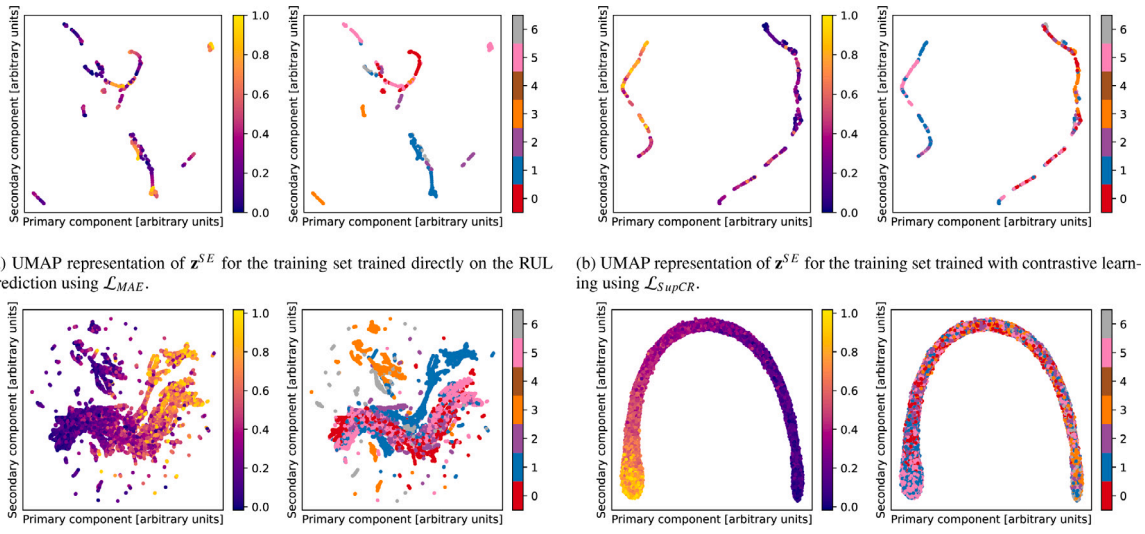
4.4.8. Contrastive learning

Contrastive learning is a semi-supervised technique that aims to learn useful representations by contrasting similar and dissimilar pairs of data samples [57,58]. In this approach, the model is trained to distinguish between positive pairs, which are similar in some way, and negative pairs, which are dissimilar. The key idea is to maximize the similarity between positive pairs while minimizing the similarity between negative pairs. This is typically achieved by training the encoded embedding vector (the output of the transformer-encoder), as depicted in Fig. 6. By optimizing the encoder to pull similar samples closer together and push dissimilar samples apart, contrastive learning enables the ISTRUST model to capture important features and patterns that are relevant to the task at hand. This process allows the model to learn generalized representations.

(a) Regular kernel weight initialization

(b) Our kernel weight initialization

Fig. 8. Custom kernel weight initialization. Fig. 8(a) demonstrates the weights of CNN kernels which are initialized by sampling from a normal distribution and allowing variance in all directions, whilst Fig. 8(b) visualizes our CNN weight initialization where no variance is allowed in the horizontal direction but the weights are still being sampled from a normal distribution.



(a) UMAP representation of $\mathbf{z}^{SE}$ for the training set trained directly on the RUL prediction using $\mathcal{L}_{MAE}$.

(b) UMAP representation of $\mathbf{z}^{SE}$ for the training set trained with contrastive learning using $\mathcal{L}_{SupCR}$.

(c) UMAP representation of $\mathbf{z}^{SE}$ for the augmented training set trained directly on the RUL prediction using $\mathcal{L}_{MAE}$.

(d) UMAP representation of $\mathbf{z}^{SE}$ for the augmented training set trained with contrastive learning using $\mathcal{L}_{SupCR}$.

Fig. 9. UMAP representation of the spatial encoded embeddings $\mathbf{z}^{SE}$. In each subfigure, the left-hand image represents the primary and secondary component of the UMAP representation of $\mathbf{z}^{SE}$, supplemented by the normalized ground truth RUL as a continuous colorbar on the right. The right-hand image also represents the primary and secondary components, supplemented by the sample number of the relevant specimen as a discrete colorbar on the right. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To visualize the embeddings obtained from the encoder, the UMAP representation is employed. The UMAP representation provides a low-dimensional visualization of the encoded embedding vector in terms of primary and secondary components. When training the whole model, i.e. both the encoder and the predictor, directly with the Mean Absolute Error (MAE) loss function $\mathcal{L}_{MAE}$, it fails to filter out information related to the specific specimen, as shown in UMAP representation in Fig. 9(a) and the specific data augmentation, as shown in Fig. 9(c). In these images, it can be seen that the spatial encoded embeddings $\mathbf{z}_{SE}$ resulting from the spatial transformer-encoder part still contain information related to the relevant specimen that is unnecessary for the RUL prediction. This means that each specimen has its own location in the UMAP representation. Consequently, the final linear projection layer can overfit to this location in the UMAP representation, and thus to the relevant specimen and/or data augmentation. Therefore, it is desired to force samples with a similar RUL to the same location in the UMAP representation, which highlights the importance of utilizing contrastive learning.

Zha et al. [56] were the first to propose contrastive learning in supervised regression problems based on the work of [57,58] in classification problems. However, they recommended the application of data augmentation at every training iteration twice, aiming to ensure convergence by always having two samples with the same RUL, also called positive samples. Because of the computational expense related to performing the data augmentation at every iteration, data augmentation is applied D times on the training set before the training process, creating the augmented training set. Using this technique, no significant difference was observed in our specific problem compared to sampling at every training iteration. Moreover, as stated previously, sampling at every iteration gives a large bottleneck in terms of computational performance. Thereby, the following loss function is used where we replaced the term $2N$ from the fundamental equation with the term N, hence:

$$\mathcal{L}_{SupCR} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{N-1}\sum_{j=1,j\neq i}^{N} log \times \frac{exp(L^2\text{-}dist(\mathbf{z}_{SE,i},\mathbf{z}_{SE,j})/\tau)}{\sum_{k=1}^{N}\mathbb{1}_{[k\neq i,\ L_{MAE}(\tilde{y}_i,\tilde{y}_k)\geq L_{MAE}(\tilde{y}_i,\tilde{y}_k)]}exp(L^2\text{-}dist(\mathbf{z}_{SE,i},\mathbf{z}_{SE,k})/\tau)} \tag{8}$$

where $\mathcal{L}_{SupCR}$ is the supervised contrastive regression loss, N is the batch size, $L^2\text{-}dist$ is the L^2 distance between the two input vectors, is the spatial encoded embedding, $\tilde{y}$ the corresponding ground truth RUL, $\mathbb{1}$ is true when the statement in brackets is satisfied and zero otherwise, τ is the temperature hyperparameter of the softmax function, and L_{MAE} is the MAE between the two input targets. Since the model fails to filter out specimen-related information (Fig. 9(a)) rather than augmentation-related information (Fig. 9(c)), sampling at every training iteration should be avoided as claimed in [56]. Note that this loss function is applied on $\mathbf{z}_{SE}$, and thus remains untrained during

the contrastive learning stage, resulting in the absence of utilizing the last layer responsible for the RUL prediction. To subsequently consider the final linear projection layer, it is trained in a second stage whilst fixing the weights of all layers obtained in the first stage via contrastive learning.

4.5. Interpretability: information flow

In the underlying section, the information flow in both the temporal and the spatial transformer-encoder will be explained by analyzing the attention weights. For simplicity, a single attention head will be assumed. At the end of the section, it will be shown that this can easily be extrapolated to the interpretable multi-head attention. Besides, since the temporal and spatial transformer-encoder work in the exact same way, only in a different domain, no distinction will be made between them during the following explanation of the information flow. The patches in the relevant domain will merely be referred to as patches, meaning that either all of them are in the temporal domain T where the spatial domain acts as a batch dimension and does not contribute to any mathematical operations in any way, or they are in the spatial domain P.

The transfer of information in the self-attention mechanism can be interpreted by analyzing the attention weights computed in self-attention and token-attention. In self-attention, the attention weights are calculated by the means of a matrix multiplication between the query $\mathbf{Q}_{\mathbf{self}}$ and key $\mathbf{K}_{\mathbf{self}}$ matrices, followed by a softmax operation. This computes the correlation between each query and all keys, representing the relationship between the patches. The resulting attention weights, represented by the matrix $\mathbf{A}_{\mathbf{self}}$, indicate how much information is transferred between patches. The actual information transfer happens when these attention weights are multiplied by the values, resulting in the intermediate self-attended patches $\mathbf{Z}'_{\mathbf{self}}$.

Similarly, in the token-attention, there is only one query originating from the learnable token. The attention weights are computed by attending the query $\mathbf{Q}_{\mathbf{token}}$ to the attended patches $\mathbf{K}_{\mathbf{token}}$ originating from the self-attention. The resulting attention weights $\mathbf{A}_{\mathbf{token}}$ determine the importance of the attended patches for the RUL prediction. Multiplying these attention weights with the values $\mathbf{V}_{\mathbf{token}}$ originating from the attended patches yields the encoded embedding.

Likewise, by tracing back the information present in the encoded embedding through the attention weights, it can accurately be determined how much information was taken from each patch to obtain the encoded embedding. Knowing that $\mathbf{V}_{\mathbf{token}}$ originates from $\mathbf{Z}'_{\mathbf{self}}$, the flow of information in the entire interpretable transformer-encoder is represented by matrix $\mathbf{A}$:

$$\mathbf{A} = \mathbf{A}_{\mathbf{token}}\mathbf{A}_{\mathbf{self}} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n_{keys}} \end{bmatrix}_{token} \times \begin{bmatrix} a_{1,1} & \cdots & a_{1,n_{keys}} \\ \vdots & \ddots & \vdots \\ a_{n_{queries},1} & \cdots & a_{n_{queries},n_{keys}} \end{bmatrix}_{self} = \begin{bmatrix} \tilde{a}_1 & \cdots & \tilde{a}_n \end{bmatrix} \tag{9}$$

where $\mathbf{A}$ represents the matrix containing how much information has been taken from each patch to arrive at the encoded embedding and $\tilde{a}_i$ represents how much information has been taken from the ith patch.

As can be seen in Fig. 1(b), the attention weights in the multi-head attention are averaged before being multiplied by the values. The same can thus be done for the attention weights in both the self-attention and the token-attention. It is important that these attention weights are averaged before applying Eq. (9) since the head i of the self-attention unmistakably has no correlation with the head i of the token-attention. It can thus be concluded that we are able to interpret the information flow of both the temporal and spatial domains separately. This explanation is achieved by reducing the attention weights in the multi-head self- and token-attention such that each patch has a single weight in the relevant domain. The resulting attention weights directly determine how much information from each patch contributes to the final prediction.

5. Experimental results and discussion

To obtain the experimental results, our ISTRUST model was trained in PyTorch in a two-stage process on an Nvidia RTX4080 16 GB GPU. Because estimating the time complexity theoretically is challenging, we approximated it by performing forward passes with different image heights, widths, frame lengths, layers, number of heads in multi-head attention, and dimensions of the encoded embeddings. The elapsed time to perform a forward pass with 1×, 2×, 4×, and 8× of the above sizes (all of them were increased simultaneously) was $0.21, 0.41, 0.96, 2.69$ s, respectively. This means that the time is increasing somewhere between linearly and quadratically with these sizes. In particular, the time increases approximately by a factor of 1.2, since $\frac{2.69}{0.96} / \frac{0.96}{0.41} = \frac{0.96}{0.41} / \frac{0.41}{0.21} \simeq 1.2$. Consequently, the time complexity is estimated approximately to be $O(n^{1.2})$.

Before evaluating the model's performance, we fine-tuned the hyperparameters by training on the training dataset and adjusted the hyperparameters according to the performance on the validation dataset. The resulting hyperparameters can be found in Table 1. During the first stage, we trained the encoder, which encompasses the entire model as outlined in Fig. 6, with the exception of the final linear projection layer. The first stage of training was conducted employing contrastive learning, utilizing $\mathcal{L}_{SupCR}$, and SGD with a momentum of 0.9 on the augmented training set. The weights were initialized according to the guidelines provided in [72]. The first stage training process was approximately 30 to 40 min per fold, depending on the rate of convergence of the specific fold in the cross-validation process.

5.1. Model performance, optimal and suboptimal RUL predictions

Through experimentation on the validation set, we discovered that the optimal encoder performance following the contrastive learning stage was attained by utilizing the encoder's state at the lowest non-augmented training loss. The resulting epochs for which the embedder states were taken can be found in Table 2. The UMAP representation of the augmented training set using contrastive learning is depicted in Fig. 9(b), accompanied by the non-augmented training set in Fig. 9(d), which was not included in the training set during the contrastive learning stage. These representations reveal that the encoder is no longer discriminating between the specimens based on the applied augmentation or the relevant specimen, with the primary variation observed in the UMAP representation relating to the ground truth RUL. This serves as the initial validation of the ISTRUST model's performance, indicating that the contrastive learning was successful since the model no longer overfits the specific specimen or augmentation and thus correctly filters out spurious information.

Following the contrastive learning stage, we froze the state of the encoder and solely trained the predictor using SGD with a momentum of 0.9 and $\mathcal{L}_{MAE}$ on the non-augmented training set for six epochs, with an observed training time of fewer than two minutes. The rationale for choosing such a small number of epochs is that the second learning stage is a much simpler process than the first stage of training. Consequently, during the hyperparameter tuning, the regression layer that exists in the second stage starts to overfit after some epochs, due to the high dimensionality of the spatial encoded embedding vector. Hence, the training is stopped after six epochs for optimal performance.

To ensure robustness in our testing, we utilized cross-validation with a total of six specimens. The RUL predictions under uncertainty utilizing MC dropout (95% CI) and the associated loss curves for specimens classified as successful can be found in Fig. 10, while Fig. 11 contains the corresponding results for specimens with suboptimal performance. Discrimination between optimal and suboptimal performances was guided by two heuristics: (i) whether RUL showed a rational decrease as it approached the EOL, and (ii) whether the spatial attention focused on damaged or noisy areas. The resulting losses can be found in Table 2. Notably, across all specimens, the RUL profile exhibited a

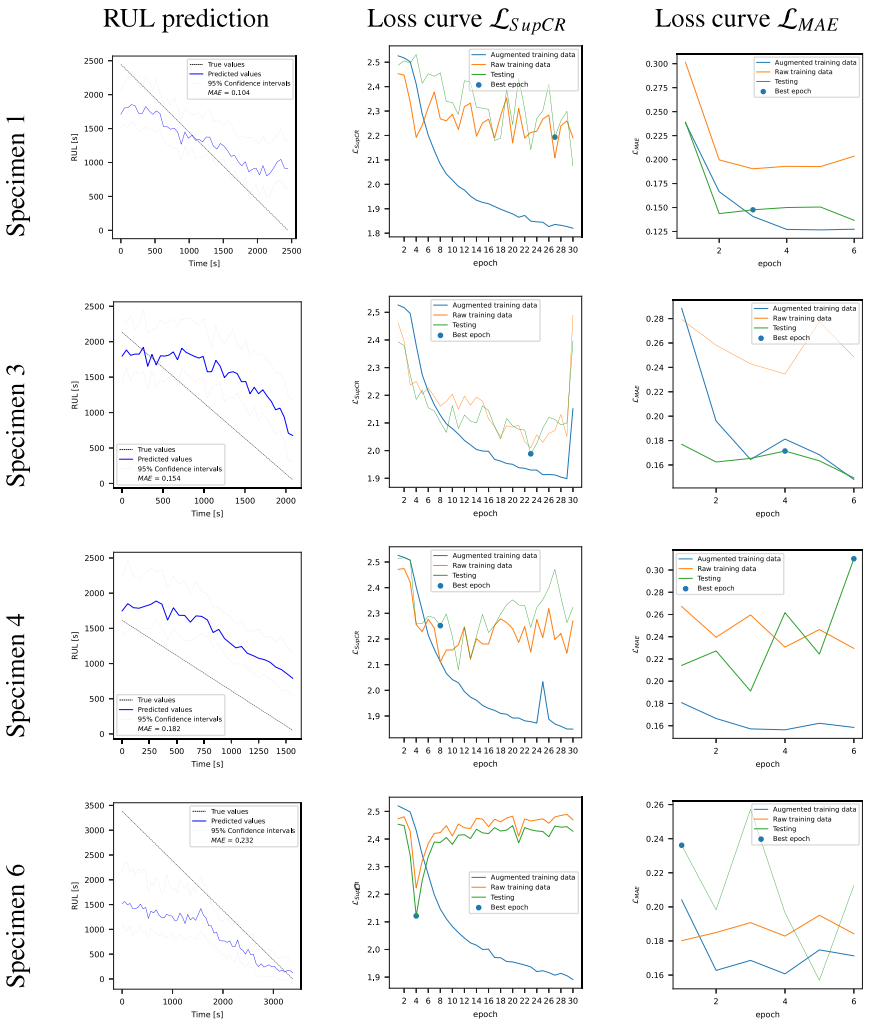


Fig. 10. Cross-validation of successful results of RUL prediction. Vertically, the table is subdivided such that each specimen has its own column. Subsequently, each row corresponds to a different fold of the cross-validation. The RUL predictions on the testing specimens can be found in the first column. In the last two columns, the loss curves with all losses are depicted.

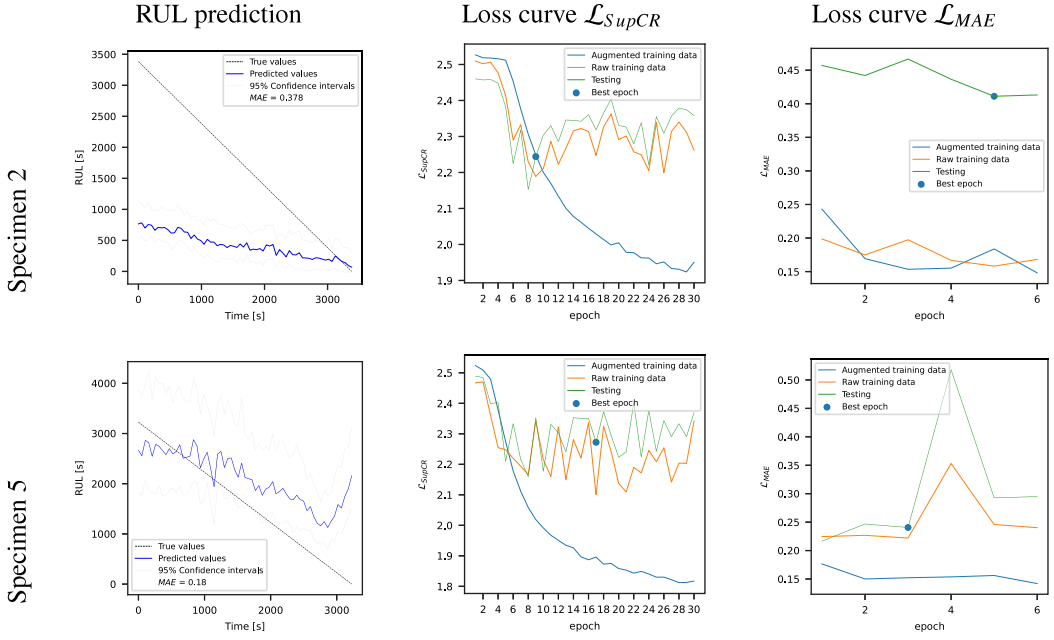


Fig. 11. Cross validation of suboptimal results of RUL prediction alongside the loss curves depicted row-wisely for each specimen.

Table 1
Values of the parameters resulting from the hyperparameter tuning on the validation set. All values are dimensionless.

Corresponding	Parameter	Values	Parameter	Values for $\mathcal{L}_{SupCR}$	Values for $\mathcal{L}_{MAE}$
Dataset	s_{skip}	2	Epochs	30	6
	s_{window}	7	Batch size	32	32
	T	3	Learning rate embedder	2.00×10^{-2}	0
	H	640	Learning rate predictor	–	1.00×10^{-3}
	W	320	Dropout model	0.3	0.3
ISTRUST model	d_{model}	96	τ	2	–
	d_k	16			
	$H_P = W_P$	16			
	P_H	40			
	P_W	20			
	P	800			
	n_{heads}	6			

Table 2
Epochs for which the encoder and predictor states were taken. On the first hand, the epochs concerning the encoder correspond to the contrastive learning stage. On the other hand, the epochs regarding the predictor represent the second training stage, i.e. the learning of the final linear projection layer. The total losses, which can be found in the rightmost column, are calculated using the corresponding specimens in each fold.

Specimen	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	Total
Epochs (encoder)	27	9	25	8	17	6	–
Epochs (predictor)	5	5	5	5	5	5	–
Training $\mathcal{L}_{MAE}$	0.13	0.15	0.15	0.16	0.14	0.17	0.15
Testing $\mathcal{L}_{MAE}$	0.14	0.41	0.15	0.31	0.29	0.21	0.26

transitional pattern, characterized by an initial phase with an almost stable, flat slope, followed by a distinct transition to a steeper slope, signifying a sudden change in the model's processing of the input. This observation alongside the optimal and suboptimal performances will be explained through the interpretation of the ISTRUST model, i.e. via the attention weights, in the following subsection. Finally, it is worth noting that the corresponding uncertainties originate from the model itself rather than the data. This explains their persistence even close to the EOL condition where they should have been neglected.

5.2. Explaining the results — model interpretation

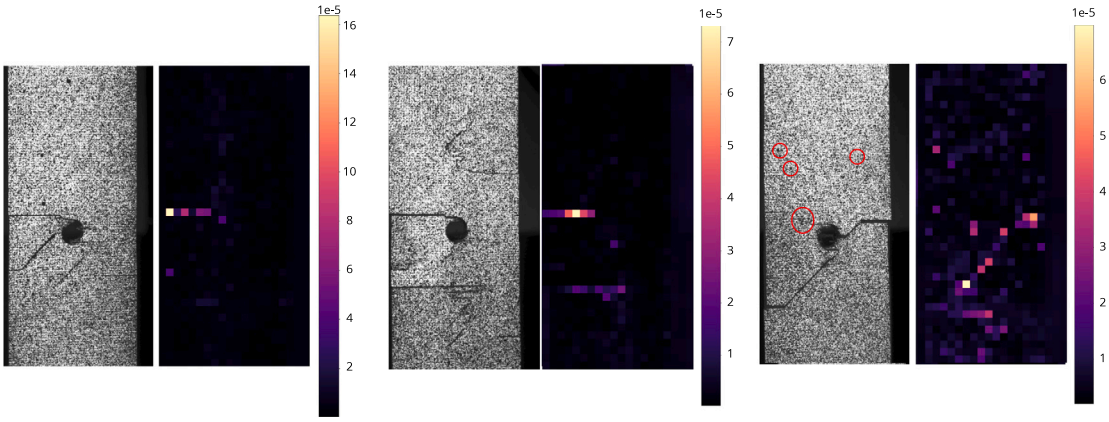
To explain the successful and suboptimal results, both the temporal and spatial attention weights originating from the multi-head self-attention and the multi-head token-attention were merged using Eq. (9), ensuring that each patch in the relevant domain has only one weight. The temporal attention weights, obtained utilizing the same equation, were further reduced by averaging over the spatial domain, with each temporal attention weight being weighted according to the corresponding spatial attention weight. It is worth noting that all the displayed attention weights correspond to testing specimens that were not part of the training sets in their respective folds.

The spatial attention weights at the EOL for three selected specimens are presented in Fig. 12. In Figs. 12(a) and 12(b), the model successfully focuses on locations with damage near the EOL, which validates the desired results for specimens 3 and 6. In Fig. 13, the spatial and temporal focus evolution of the ISTRUST model throughout the entire life of specimens 4 and 5 are depicted. In Fig. 13(a), it is evident that initially, the model rationally fails to capture the minor damage present in the specimens and instead focuses on spurious parts at random locations as the damage still remains indistinguishable. Consequently, the initial RUL values for each specimen exhibit a relatively consistent range, indicating that the model primarily relies on the average RUL derived from the entire dataset during its initial predictions. Nevertheless, as the damage accumulates, delamination starts being detected by the model successfully, which focuses only on the important parts of the image, leading to an accurate decrease in the predicted RUL. Similar observations are present for specimens 1, 3, and 6 shown in Fig. A.1. For these specimens there was significant delamination present, resulting in a successful RUL prediction. Furthermore, an examination of the temporal attention weights in the same figure reveals that the ISTRUST model reasonably prioritizes the latter images over the earlier ones. Nevertheless, the model does not entirely discard the initial ones. This can be attributed to the model's need for the earlier images to estimate the speed of damage accumulation while relying on the last image to assess the severity of damage in the current state.

In order to explain the suboptimal results of specimens 2 and 5, the spatial attention weights of specimen 5 at the EOL are shown in Fig. 12(c). It can be acknowledged that for this specimen, the model also captures the cracks, despite the less prevalent damage and the relatively poorer RUL predictions. This can be attributed to two reasons. Firstly, due to the attention weights that focus on some spurious parts, circled in red. These are typically black dots, which distract the spatial attention from the actual damage. In this case, the model gratuitously focuses on additional parts of the image, which confuses its RUL estimation. This can be efficiently seen in Fig. 13(b) in the last image representing a sample close to the EOL, where the model additionally considers some spurious parts as damage, thus violently changing its RUL estimation. Secondly, for specimen 5 specifically, significant cracks are observed at a 45-degree angle, a phenomenon not present in the other specimens. While the model does detect these cracks, it does not effectively correlate them with a reduction in the RUL curve. This is because similar cracks were not encountered during the training phase, leading to an unexpected shift in the RUL trajectory. Nevertheless, the RUL predictions in Fig. 11 still exhibit a negative slope in general (with the only exception being the specimen's 5 latest RUL predictions), specifying that the ISTRUST model indeed captured, but underestimated the extent of the damage. Similar observations can be noted for specimen 2 in Fig. A.1(b). Consequently, it can be concluded that there is a noticeable correlation between the accuracy of the spatial attention weights, the severity of the visible damage, and the accuracy of the RUL prediction.

It should be acknowledged that even though the model managed to capture the majority of the damage, it only captures the corresponding one that is relevant for the RUL prediction. As a result, *the proposed ISTRUST model can currently not be accurately used as an anomaly detection method.* However, the shown results are promising, and it should be further investigated whether the attention weights can be leveraged for anomaly detection by modifying the training setup, the size of the dataset, or the model's architecture.

Lastly, the proposed model encounters the primary limitations associated with the DIC technique and the camera systems. Regarding DIC analysis, the primary limitation encompasses the extended computational time required for post-processing DIC data to extract strain fields, which hinders the application of this technique in real-time scenarios. Even upon extracting the strain field, post-processing fails to accurately identify high-damage areas due to the inability to measure substantial deformations and the corresponding strains [73]. Furthermore, the

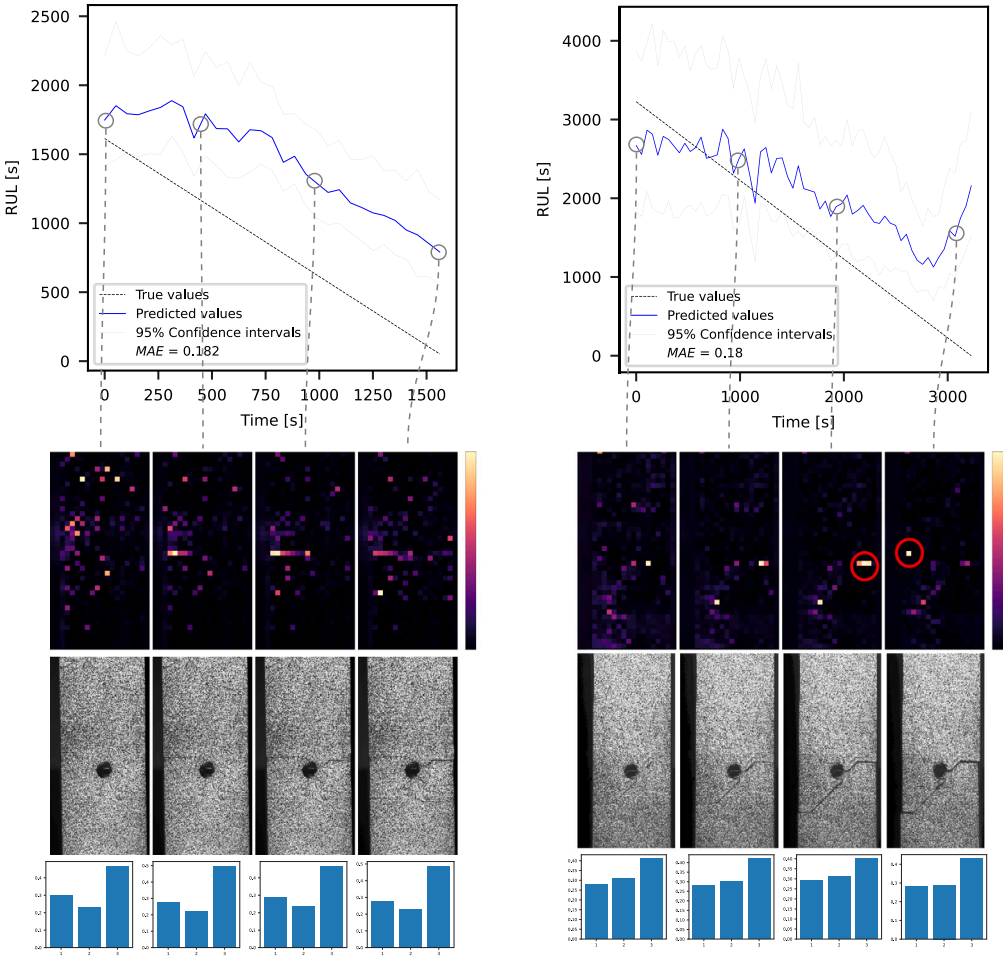


(a) Spatial attention map of selected successful results for specimen 3

(b) Spatial attention map of selected successful results for specimen 6

(c) Spatial attention map of selected suboptimal results for specimen 5. The red circles indicate artifacts like black dots correlating to the spatial attention weights.

Fig. 12. Spatial attention weights of selected testing specimens at the EOL condition. The displayed colormaps indicate the spatial attention weights, accompanied by the input image on their left. The color within the colormap, along with the accompanying colorbar on the right, indicates the absolute magnitude of the attention weights. In Figs. 12(a) and 12(b) the spatial attention weights of specimens 2, 5 for which the results were deemed successful are shown. In Fig. 12(c) the spatial attention weights of specimen 3 which is classified as suboptimal are depicted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Specimen 4

(b) Specimen 5

Fig. 13. Evolution of spatial and the corresponding temporal attention weights of testing specimens four (Fig. 13(a)) and five (Fig. 13(b)). Each figure consists of three parts. The top graph represents the predicted RUL for each specimen. The colormaps displayed beneath the graph illustrate the changes in spatial attention weights over time. Additionally, for each attention map, the last input image can be found below each attention map. The color within the colormap, along with the accompanying colorbar on the right, indicates the relative magnitude of the attention weights. The bottom graph corresponds to the temporal attention weights of each sequence of images (here, only the last and most important image of each sequence is shown). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

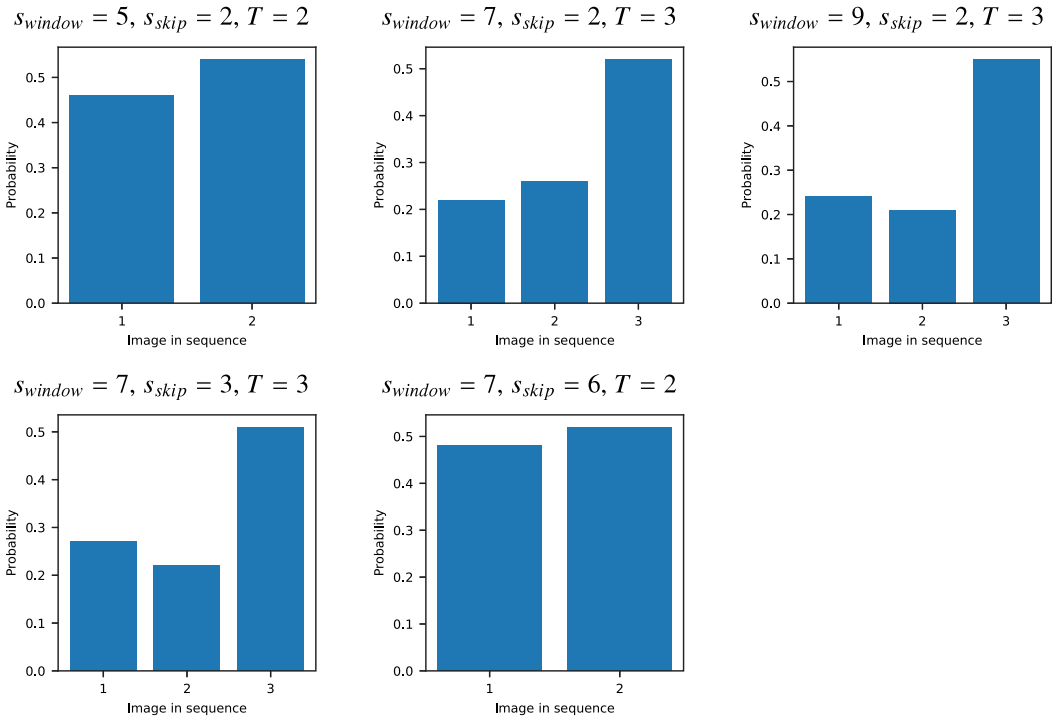


Fig. 14. Temporal attention weights when testing specimen 1 with varying hyperparameter values that affect the time domain.

quality of the speckle pattern applied to the surface and how well it is captured by the camera can limit the feature extraction accuracy. Applying the model directly on raw images mitigates the risk that an inappropriate speckle pattern may introduce.

5.3. The role of temporal attention

The role of spatial attention is well understood; however, the interest in temporal attention is less evident. To address this, a parametric study was conducted focusing on the key hyperparameters associated with temporal attention, specifically skip size (s_{skip}) and window size (s_{window}). Due to extensive memory requirements, the values were selected to ensure that the number of images (T) did not exceed three. Fig. 14 illustrates examples of temporal attention weights and their behavior with varying hyperparameter values when testing specimen 1. Notably, when $T = 3$, the final image in the sequence is consistently more significant than the preceding images for predicting RUL. Conversely, when $T = 2$, both images hold nearly equal importance for RUL prediction. This observation can be attributed to the fact that a sequence of at least two images allows the model to capture the crack propagation speed, which is closely related to RUL. This finding underscores the importance of temporal attention weights and utilizing sequential images rather than a single image as input.

Finally, for the sake of comprehensiveness, we present the impact of these hyperparameters on prognostics in Table 3. It is evident that the prognostics are significantly influenced by these hyperparameters. When more images are skipped, the model's efficiency drops significantly. Having small or large window sizes negatively affects the model's performance as well.

6. Conclusions and recommendations

In this paper, a novel model architecture – namely the ISTRUST model – based on vision transformers is proposed capable of predicting the RUL under uncertainty, given sequences of raw images as input, with a primary focus on interpretability. This model is evaluated on an experimental dataset acquired from composite samples that are under fatigue loads and visible cracks as damage propagates. To interpret the correlation between the input images taken from two cameras and the RUL predictions an innovative attention mechanism is proposed based on the decomposition of the spatiotemporal domain. By separating the temporal and spatial domains and leveraging the attention mechanism, the black-box commonly associated with deep learning architectures is circumvented, allowing for an interpretable AI prognostic model. The spatial and temporal attention weights demonstrated the model's ability to correctly prioritize patches with higher levels of damage in the spatial and temporal domain respectively. Besides, a noticeable correlation was observed between the accuracy of the spatial attention weights and the accuracy of the RUL prediction. Based on that, it was shown that despite the RUL prediction being initially a flat line for all specimens, as soon as the model focused on the damage, it started to drop at the anticipated negative slope. As predictions depend entirely on raw data inputs, the ISTRUST model has the ability to identify situations where predictions may fall short or perform well. Consequently, it is reasonable to expect that the model's performance depends on the quality of the data provided. Nevertheless, its primary goal is to provide insights and understanding about the accuracy of predictions, aiming to achieve an exceptional level of reliability and interpretation, unfolding the barriers of the black-box models.

Because of the limited data acquired from the experiment, a data augmentation technique is performed, thus increasing the risk of overfitting. In this regard, contrastive learning is utilized to help the ISTRUST model distinguish the important information responsible for crack propagation, which in turn affects the RUL, and filters out the spurious one. The UMAP representation is responsible for visualizing the results of the contrastive learning. Furthermore, the stochasticity of the RUL is naturally included in our ISTRUST model via the

Table 3
Parametric study of the hyperparameters s_{window} and s_{skip} and their effects on prognostics using the corresponding specimens in each fold.

Examined values	Specimen	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	Total
$s_{window} = 5,\ s_{skip} = 2,\ T = 2$	Epochs (encoder)	27	17	17	16	22	28	–
	Epochs (predictor)	6	2	4	5	5	5	–
	Training $\mathcal{L}_{MAE}$	0.17	0.14	0.17	0.20	0.20	0.21	0.18
	Testing $\mathcal{L}_{MAE}$	0.16	0.42	0.19	0.32	0.30	0.31	0.28
$\mathbf{s_{window} = 7,\ s_{skip} = 2,\ T = 3}$	Epochs (encoder)	27	9	25	8	17	6	–
	Epochs (predictor)	5	5	5	5	5	5	–
	Training $\mathcal{L}_{MAE}$	0.13	0.15	0.15	0.16	0.14	0.17	**0.15**
	Testing $\mathcal{L}_{MAE}$	0.14	0.41	0.15	0.31	0.29	0.21	**0.26**
$s_{window} = 9,\ s_{skip} = 2,\ T = 3$	Epochs (encoder)	27	23	20	12	27	7	–
	Epochs (predictor)	3	2	6	1	5	6	–
	Training $\mathcal{L}_{MAE}$	0.25	0.18	0.21	0.22	0.20	0.23	0.22
	Testing $\mathcal{L}_{MAE}$	0.29	0.42	0.20	0.32	0.29	0.21	0.28
$s_{window} = 7,\ s_{skip} = 3,\ T = 3$	Epochs (predictor)	5	3	5	2	5	6	–
	Training $\mathcal{L}_{MAE}$	0.24	0.17	0.19	0.18	0.20	0.21	0.20
	Testing $\mathcal{L}_{MAE}$	0.25	0.43	0.17	0.31	0.32	0.31	0.30
$s_{window} = 7,\ s_{skip} = 6,\ T = 2$	Epochs (predictor)	3	2	6	1	5	6	–
	Training $\mathcal{L}_{MAE}$	0.28	0.22	0.26	0.27	0.23	0.28	0.26
	Testing $\mathcal{L}_{MAE}$	0.26	0.47	0.28	0.34	0.31	0.31	0.33

MC dropout, offering a simple yet meaningful representation of the introduced uncertainty.

Despite the weight initialization being designed to assist the model in capturing horizontal cracks, it notably succeeded in identifying diagonal cracks during the learning process. As a result, the model demonstrates the potential for generalization to more complex crack shapes. However, in such scenarios, further testing with various weight initializations is necessary to determine the optimal configuration. It is imperative to acknowledge that the naturally occurring cracks within the internal sections of the structure are detectable by the cameras only when they are detected on the surface. Consequently, in scenarios where the specimens are black, such as with carbon fiber composites, cracks may be difficult to detect visually. In these cases, it would be advisable to apply a white painted surface to the specimens before using the model to enhance crack visibility and ensure accurate analysis. This condition is critical for the application of this model and warrants further investigation in the future.

The objective of this work is to develop a generalizable model applicable to any structure from which raw sequential images can be obtained. While the current task focused on predicting RUL from these images to demonstrate the model's capability in transforming high-dimensional data into straightforward one-dimensional estimations under uncertainty, this approach can be extended to other tasks such as crack segmentation or classification. This can be achieved by modifying the final FC layers of the proposed architecture and updating the loss function accordingly.

Although the model's training was performed offline, it can be easily considered for real-time applications by positioning one or more cameras to the examined in-service structure. Nevertheless, switching from in-lab tests to real-time prognostics involves technical challenges like ensuring data quality, model accuracy, computational efficiency, and operational challenges related to system integration, maintenance, and network constraints. This is a promising direction for further extending this work towards real-time prognostics.

One key limitation of the present work is the lack of comprehensive data that captures the full range of failure mechanisms inherent to composite materials. The fatigue life of composites is highly dependent on the initiation, interaction, and propagation of cracks, which are intrinsically stochastic and can lead to significant variability in RUL predictions. In particular, specimen 5, with its major cracks oriented at 45 degrees, highlights the need for training data that includes diverse failure modes. Additional future work should address this by incorporating a wider variety of damage scenarios to improve the robustness of the model.

While it is recognized that the ISTRUST model's predictive performance may not be ideal for specific cases, it is essential to highlight that our approach provides a logical and cohesive explanation for the underlying factors influencing this observation. This lays the foundation for creative and forward-thinking ideas to enhance its effectiveness in the future. Additionally, it is recommended to apply our proposed architecture to larger datasets, hence the resolution of the model could be increased without the potential of overfitting. This is because higher resolution could avoid the distraction of the spatial attention weights to spurious features and allow the model to capture less extensive damage, thus further increasing the model's performance. Finally, because of the accurate spatial attention maps, it should be further investigated whether the attention weights can be leveraged for anomaly detection by modifying the training setup, the size of the dataset, or the model architecture.

Code availability

All code was implemented in Python using PyTorch as the primary DL package. All code and scripts to reproduce the experiments of this paper are available at https://github.com/Center-of-Excellence-AI-for-Structures/ISTRUST_MODEL.

Fundings

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

P. Komninos: Writing – review & editing, Writing – original draft, Software, Methodology, Data curation, Conceptualization. **A.E.C. Verraest:** Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis. **N. Eleftheroglou:** Writing – review & editing, Resources, Investigation. **D. Zarouchas:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

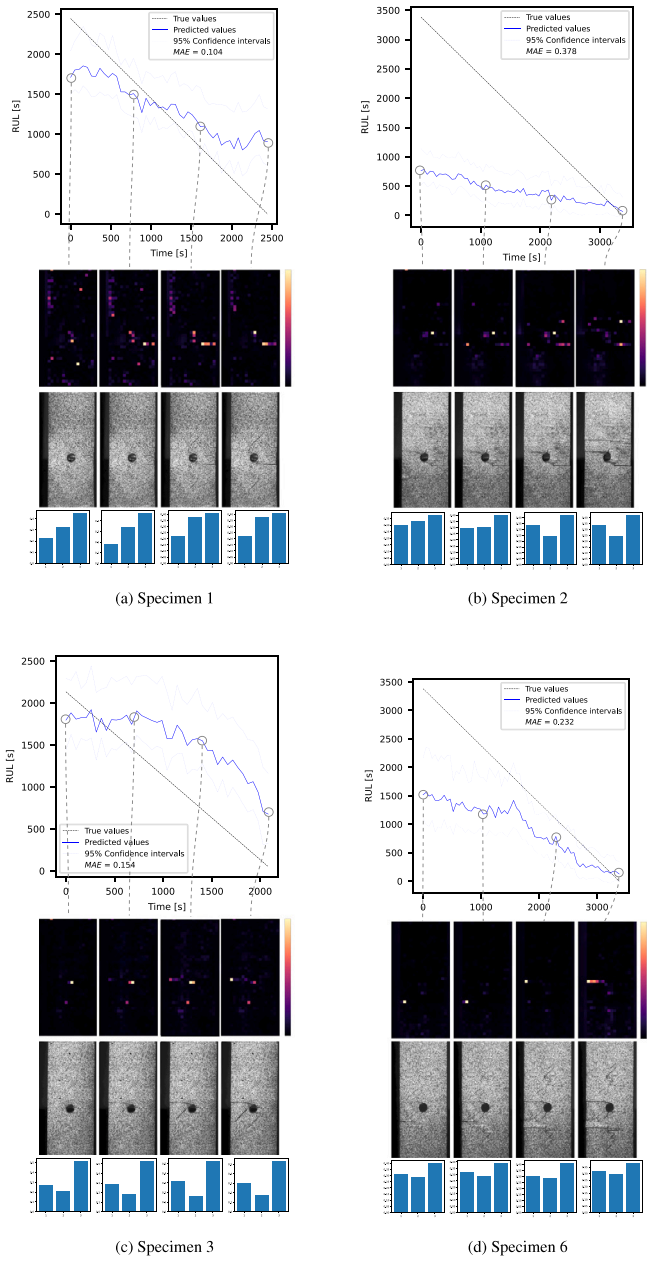
The dataset used for this work is available at https://data.mendeley.com/datasets/ky3gb8rk9h/1.

Declaration of Generative AI in scientific writing

During the preparation of this work, the authors used ChatGPT based on GPT3.5 in order to improve the readability and language of some parts of the paper. The tool was in no way used to analyze and draw insights from the data, perform literature research, or extract any information other than feedback on the writing style based on the provided inputs. The tool was only used to perform minimal changes and provide feedback based on the provided input text, where the scientific content of the input sentences remains unchanged. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

Appendix. Additional illustrations of the attention evolution

See Fig. A.1.



(a) Specimen 1 (b) Specimen 2

(c) Specimen 3 (d) Specimen 6

Fig. A.1. Evolution of spatial and the corresponding temporal attention weights of the remaining testing specimens.

References

[1] Xia Tangbin, Dong Yifan, Xiao Lei, Du Shichang, Pan Ershun, Xi Lifeng. Recent advances in prognostics and health management for advanced manufacturing paradigms. Reliab Eng Syst Saf 2018;178:255–68. http://dx.doi.org/10.1016/j.ress.2018.06.021, URL https://www.sciencedirect.com/science/article/pii/S095183201731459X.

[2] Lee Jay, Wu Fangji, Zhao Wenyu, Ghaffari Mahsa, Liao Linxia, Siegel David. Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. Mech Syst Signal Process 2014;42:314–34. http://dx.doi.org/10.1016/j.ymssp.2013.06.004.

[3] Li Hongbo, Zheng Wenli, Tang Feilong, Zhu Yanmin, Huang Jielong. Few-shot time-series anomaly detection with unsupervised domain adaptation. Inform Sci 2023;649:119610. http://dx.doi.org/10.1016/j.ins.2023.119610, URL https://www.sciencedirect.com/science/article/pii/S0020025523011957.

[4] Liu Chang, Yuan Zhong, Chen Baiyang, Chen Hongmei, Peng Dezhong. Fuzzy granular anomaly detection using Markov random walk. Inform Sci 2023;646:119400. http://dx.doi.org/10.1016/j.ins.2023.119400, URL https://www.sciencedirect.com/science/article/pii/S0020025523009854.

[5] Yan Lejing, Luo Chao, Shao Rui. Discrete log anomaly detection: A novel time-aware graph-based link prediction approach. Inform Sci 2023;647:119576. http://dx.doi.org/10.1016/j.ins.2023.119576, URL https://www.sciencedirect.com/science/article/pii/S0020025523011611.

[6] Wang Zhuqing, Liu Ning, Chen Chilian, Guo Yangming. Adaptive self-attention LSTM for RUL prediction of lithium-ion batteries. Inform Sci 2023;635:398–413. http://dx.doi.org/10.1016/j.ins.2023.01.100, URL https://www.sciencedirect.com/science/article/pii/S0020025523001007.

[7] Zhou Chunjie, Hou Aihua, Dai Pengfei, Li Ali, Zhang Zhenxing, Mu Yuejun, Liu Li. Risk factor refinement and ensemble deep learning methods on prediction of heart failure using real healthcare records. Inform Sci 2023;637:118932. http://dx.doi.org/10.1016/j.ins.2023.04.011, URL https://www.sciencedirect.com/science/article/pii/S0020025523004991.

[8] Yan Aijun, Wang Weixian, Zhang Chunxiao, Zhao Hui. A fault prediction method that uses improved case-based reasoning to continuously predict the status of a shaft furnace. Inform Sci 2014;259:269–81. http://dx.doi.org/10.1016/j.ins.2013.04.025, URL https://www.sciencedirect.com/science/article/pii/S0020025513003290.

[9] Hao Shengang, Zheng Jun, Yang Jie, Sun Haipeng, Zhang Quanxin, Zhang Li, Jiang Nan, Li Yuanzhang. Deep reinforce learning for joint optimization of condition-based maintenance and spare ordering. Inform Sci 2023;634:85–100. http://dx.doi.org/10.1016/j.ins.2023.03.064, URL https://www.sciencedirect.com/science/article/pii/S0020025523003572.

[10] Baraldi Piero, Mangili Francesca, Zio Enrico. A belief function theory based approach to combining different representation of uncertainty in prognostics. Inform Sci 2015;303:134–49. http://dx.doi.org/10.1016/j.ins.2014.12.051, URL https://www.sciencedirect.com/science/article/pii/S0020025515000031.

[11] Hu Yang, Miao Xuewen, Si Yong, Pan Ershun, Zio Enrico. Prognostics and health management: A review from the perspectives of design, development and decision. Reliab Eng Syst Saf 2022;217:108063. http://dx.doi.org/10.1016/j.ress.2021.108063, URL https://www.sciencedirect.com/science/article/pii/S0951832021005652.

[12] Thomopoulos Rallou, Destercke Sébastien, Charnomordic Brigitte, Johnson Iyan, Abécassis Joël. An iterative approach to build relevant ontology-aware data-driven models. Inform Sci 2013;221:452–72. http://dx.doi.org/10.1016/j.ins.2012.09.015, URL https://www.sciencedirect.com/science/article/pii/S0020025512006081.

[13] Li Tongyang, Wang Shaoping, Zio Enrico, Shi Jian, Ma Zhonghai. A numerical approach for predicting the remaining useful life of an aviation hydraulic pump based on monitoring abrasive debris generation. Mech Syst Signal Process 2020;136:106519. http://dx.doi.org/10.1016/j.ymssp.2019.106519, URL https://www.sciencedirect.com/science/article/pii/S088832701930740X.

[14] Xu Xingwei, Li Xiang, Ming Weiwei, Chen Ming. A novel multi-scale CNN and attention mechanism method with multi-sensor signal for remaining useful life prediction. Comput Ind Eng 2022;169:108204. http://dx.doi.org/10.1016/j.cie.2022.108204, URL https://www.sciencedirect.com/science/article/pii/S0360835222002741.

[15] Huang Cheng-Geng, Huang Hong-Zhong, Li Yan-Feng, Peng Weiwen. A novel deep convolutional neural network-bootstrap integrated method for RUL prediction of rolling bearing. J Manuf Syst 2021;61:757–72. http://dx.doi.org/10.1016/j.jmsy.2021.03.012, URL https://www.sciencedirect.com/science/article/pii/S0278612521000674.

[16] Yang Yong, Chen Hongmei, Mi Yong, Luo Chuan, Horng Shi-Jinn, Li Tianrui. Multi-label feature selection based on stable label relevance and label-specific features. Inform Sci 2023;648:119525. http://dx.doi.org/10.1016/j.ins.2023.119525, URL https://www.sciencedirect.com/science/article/pii/S0020025523011106.

[17] Eleftheroglou Nick, Loutas Theodoros. Fatigue damage diagnostics and prognostics of composites utilizing structural health monitoring data and stochastic processes. Struct Health Monit 2016;15(4):473–88. http://dx.doi.org/10.1177/1475921716646579, arXiv:https://doi.org/10.1177/1475921716646579.

[18] Behera Sourajit, Misra Rajiv, Sillitti Alberto. Multiscale deep bidirectional gated recurrent neural networks based prognostic method for complex non-linear degradation systems. Inform Sci 2021;554:120–44. http://dx.doi.org/10.1016/j.ins.2020.12.032, URL https://www.sciencedirect.com/science/article/pii/S0020025520311981.

[19] Deutsch Jason, He David. Using deep learning-based approach to predict remaining useful life of rotating components. IEEE Trans Syst Man Cybern 2018;48(1):11–20. http://dx.doi.org/10.1109/TSMC.2017.2697842.

[20] Wang Yuyan, Wang Dujuan, Ye Xin, Wang Yanzhang, Yin Yunqiang, Jin Yaochu. A tree ensemble-based two-stage model for advanced-stage colorectal cancer survival prediction. Inform Sci 2019;474:106–24. http://dx.doi.org/10.1016/j.ins.2018.09.046, URL https://www.sciencedirect.com/science/article/pii/S002002551830759X.

[21] Eleutheroglou Nick, Zarouchas Dimitrios, Benedictus Rinze. An adaptive probabilistic data-driven methodology for prognosis of the fatigue life of composite structures. Compos Struct 2020;245:112386. http://dx.doi.org/10.1016/j.compstruct.2020.112386, URL https://www.sciencedirect.com/science/article/pii/S0263822319347634.

[22] Eleutheroglou Nick, Galanopoulos Georgios, Loutas Theodoros. Similarity learning hidden semi-Markov model for adaptive prognostics of composite structures. Reliab Eng Syst Saf 2024;243:109808. http://dx.doi.org/10.1016/j.ress.2023.109808, URL https://www.sciencedirect.com/science/article/pii/S0951832023007226.

[23] Akrim Anass, Gogu Christian, Nerville Thomas Guillebot de, Strähle Paul, Pagou Brondon Waffa, Salaün Michel, Vingerhoeds Rob. A framework for generating large data sets for fatigue damage prognostic problems. In: 2022 IEEE international conference on prognostics and health management. ICPHM, 2022, p. 25–33. http://dx.doi.org/10.1109/ICPHM53196.2022.9815692.

[24] Akrim Anass, Gogu Christian, Vingerhoeds Rob, Salaün Michel. Self-supervised learning for data scarcity in a fatigue damage prognostic problem. Eng Appl Artif Intell 2023;120:105837. http://dx.doi.org/10.1016/j.engappai.2023.105837, URL https://www.sciencedirect.com/science/article/pii/S0952197623000210.

[25] Nguyen Tuan-Khai, Ahmad Zahoor, Kim Jong-Myon. A deep-learning-based health indicator constructor using Kullback–Leibler divergence for predicting the remaining useful life of concrete structures. Sensors 2022;22(10). http://dx.doi.org/10.3390/s22103687, URL https://www.mdpi.com/1424-8220/22/10/3687.

[26] Zhao Chengying, Huang Xianzhen, Li Yuxiong, Yousaf Iqbal Muhammad. A double-channel hybrid deep neural network based on CNN and BiLSTM for remaining useful life prediction. Sensors 2020;20(24). http://dx.doi.org/10.3390/s20247109, URL https://www.mdpi.com/1424-8220/20/24/7109.

[27] Zhou Yexu, Hefenbrock Michael, Huang Yiran, Riedel Till, Beigl Michael. Automatic remaining useful life estimation framework with embedded convolutional LSTM as the backbone. In: Dong Yuxiao, Mladenić Dunja, Saunders Craig, editors. Machine learning and knowledge discovery in databases: applied data science track. Cham: Springer International Publishing; 2021, p. 461–77.

[28] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N, Kaiser Łukasz, Polosukhin Illia. Attention is all you need. In: Guyon I, Luxburg U Von, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. Advances in neural information processing systems. Vol. 30, Curran Associates, Inc.; 2017, URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[29] Dosovitskiy Alexey, Beyer Lucas, Kolesnikov Alexander, Weissenborn Dirk, Zhai Xiaohua, Unterthiner Thomas, Dehghani Mostafa, Minderer Matthias, Heigold Georg, Gelly Sylvain, Uszkoreit Jakob, Houlsby Neil. An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th international conference on learning representations, ICLR 2021, virtual event, Austria, May 3-7, 2021. OpenReview.net; 2021, URL https://openreview.net/forum?id=YicbFdNTTy.

[30] Bertasius Gedas, Wang Heng, Torresani Lorenzo. Is space-time attention all you need for video understanding? In: Meila Marina, Zhang Tong, editors. Proceedings of the 38th international conference on machine learning. Proceedings of machine learning research, vol. 139, PMLR; 2021, p. 813–24, URL https://proceedings.mlr.press/v139/bertasius21a.html.

[31] Plizzari Chiara, Cannici Marco, Matteucci Matteo. Spatial temporal transformer network for skeleton-based action recognition. In: Del Bimbo Alberto, Cucchiara Rita, Sclaroff Stan, Farinella Giovanni Maria, Mei Tao, Bertini Marco, Escalante Hugo Jair, Vezzani Roberto, editors. Pattern recognition. ICPR international workshops and challenges. Cham: Springer International Publishing; 2021, p. 694–701.

[32] Arnab Anurag, Dehghani Mostafa, Heigold Georg, Sun Chen, Lucic Mario, Schmid Cordelia. ViViT: A video vision transformer. 2021, CoRR abs/2103.15691. URL https://arxiv.org/abs/2103.15691.

[33] Li Xinyao, Li Jingjing, Zuo Lin, Zhu Lei, Shen Heng Tao. Domain adaptive remaining useful life prediction with transformer. IEEE Trans Instrum Meas 2022;71:1–13. http://dx.doi.org/10.1109/TIM.2022.3200667.

[34] Zhang Zhizheng, Song Wen, Li Qiqiang. Dual-aspect self-attention based on transformer for remaining useful life prediction. IEEE Trans Instrum Meas 2022;71:1–11. http://dx.doi.org/10.1109/TIM.2022.3160561.

[35] Chen Daoquan, Hong Weicong, Zhou Xiuze. Transformer network for remaining useful life prediction of lithium-ion batteries. IEEE Access 2022;10:19621–8. http://dx.doi.org/10.1109/ACCESS.2022.3151975.

[36] Wahid Abdul, Yahya Muhammad, Breslin John G, Intizar Muhammad Ali. Self-attention transformer-based architecture for remaining useful life estimation of complex machines. Procedia Comput Sci 2023;217:456–64. http://dx.doi.org/10.1016/j.procs.2022.12.241, URL https://www.sciencedirect.com/science/article/pii/S1877050922023195. 4th International Conference on Industry 4.0 and Smart Manufacturing.

[37] Li Qinghua, Yang Ying. Transformer model for remaining useful life prediction of aeroengine. J Phys Conf Ser 2022;2171(1):012072. http://dx.doi.org/10.1088/1742-6596/2171/1/012072.

[38] Mo Y, Wu Q, Li X, et al. Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit. J Intell Manuf 2021;32(8):1997–2006. http://dx.doi.org/10.1007/s10845-021-01750-x.

[39] Guo Jie, Nie Xiushan, Ma Yuling, Shaheed Kashif, Ullah Inam, Yin Yilong. Attention based consistent semantic learning for micro-video scene recognition. Inform Sci 2021;543:504–16. http://dx.doi.org/10.1016/j.ins.2020.05.064, URL https://www.sciencedirect.com/science/article/pii/S0020025520304758.

[40] Wei Longsheng, Zong Guanyu. EGA-net: Edge feature enhancement and global information attention network for RGB-D salient object detection. Inform Sci 2023;626:223–48. http://dx.doi.org/10.1016/j.ins.2023.01.032, URL https://www.sciencedirect.com/science/article/pii/S0020025523000324.

[41] Hua Cam-Hao, Huynh-The Thien, Bae Sung-Ho, Lee Sungyoung. Cross-attentional bracket-shaped convolutional network for semantic image segmentation. Inform Sci 2020;539:277–94. http://dx.doi.org/10.1016/j.ins.2020.06.023, URL https://www.sciencedirect.com/science/article/pii/S0020025520306101.

[42] Liao Shiyun, Liu Huijun, Yang Jianxi, Ge Yongxin. A channel-spatial-temporal attention-based network for vibration-based damage detection. Inform Sci 2022;606:213–29. http://dx.doi.org/10.1016/j.ins.2022.05.042, URL https://www.sciencedirect.com/science/article/pii/S0020025522004686.

[43] Kim Jinkyu, Canny John. Interpretable learning for self-driving cars by visualizing causal attention. 2017, arXiv:1703.10631.

[44] Boukhtache S, Abdelouahab K, Berry F, Blaysat B, Grédiac M, Sur F. When deep learning meets digital image correlation. Opt Lasers Eng 2021;136:106308. http://dx.doi.org/10.1016/j.optlaseng.2020.106308, URL https://www.sciencedirect.com/science/article/pii/S0143816620306588.

[45] Wang Yin, Zhao Jiaqing. DIC-net: Upgrade the performance of traditional DIC with Hermite dataset and convolution neural network. Opt Lasers Eng 2023;160:107278. http://dx.doi.org/10.1016/j.optlaseng.2022.107278, URL https://www.sciencedirect.com/science/article/pii/S0143816622003311.

[46] Cheng X, Zhou S, Xing T, Zhu Y, Ma S. Solving digital image correlation with neural networks constrained by strain-displacement relations. Opt Express 2023;31(3):3865–80. http://dx.doi.org/10.1364/OE.475232.

[47] Pantoja-Rosero BG, Oner D, Kozinski M, Achanta R, Fua P, Perez-Cruz F, Beyer K. TOPO-loss for continuity-preserving crack detection using deep learning. Constr Build Mater 2022;344:128264. http://dx.doi.org/10.1016/j.conbuildmat.2022.128264, URL https://www.sciencedirect.com/science/article/pii/S0950061822019250.

[48] Dais Dimitris, Bal İhsan Engin, Smyrou Eleni, Sarhosis Vasilis. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. Autom Constr 2021;125:103606. http://dx.doi.org/10.1016/j.autcon.2021.103606, URL https://www.sciencedirect.com/science/article/pii/S0926580521000571.

[49] Kim In-Ho, Jeon Haemin, Baek Seung-Chan, Hong Won-Hwa, Jung Hyung-Jo. Application of crack identification techniques for an aging concrete bridge inspection using an unmanned aerial vehicle. Sensors 2018;18(6). http://dx.doi.org/10.3390/s18061881, URL https://www.mdpi.com/1424-8220/18/6/1881.

[50] Khani Mahtab Mohtasham, Vahidnia Sahand, Ghasemzadeh Leila, Ozturk Y Eren, Yuvalaklioglu Mustafa, Akin Selim, Ure Nazim Kemal. Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines. Struct Health Monit 2020;19(5):1440–52. http://dx.doi.org/10.1177/1475921719883202, arXiv:https://doi.org/10.1177/1475921719883202.

[51] Zhang Allen, Wang Kelvin CP, Fei Yue, Liu Yang, Tao Siyu, Chen Cheng, Li Joshua Q, Li Baoxian. Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet. J Comput Civ Eng 2018;32(5):04018041. http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000775, URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CP.1943-5487.0000775.

[52] Zhu Shun-Peng, Wang Lanyi, Luo Changqi, Correia Jos, Jesus Ablio, Berto Filippo, Wang Qingyuan. Physics-informed machine learning and its structural integrity applications: state of the art. Philos Trans R Soc Lond Ser A Math Phys Eng Sci 2023;381. http://dx.doi.org/10.1098/rsta.2022.0406.

[53] Song Lu-Kai, Li Xue-Qin, Zhu Shun-Peng, Choy Yat-Sze. Cascade ensemble learning for multi-level reliability evaluation. Aerosp Sci Technol 2024;148:109101. http://dx.doi.org/10.1016/j.ast.2024.109101, URL https://www.sciencedirect.com/science/article/pii/S1270963824002347.

[54] Eleutheroglou Nick. Adaptive prognostics: a reliable RUL approach. In: Proceedings of the annual conference of the PHM society 2023. Annual Conference of the PHM Society; 2023, http://dx.doi.org/10.36001/phmconf.2023.v15i1.3495, URL https://papers.phmsociety.org/index.php/phmconf/article/view/3495.

[55] Gal Yarin, Ghahramani Zoubin. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Balcan Maria Florina, Weinberger Kilian Q, editors. Proceedings of the 33rd international conference on machine learning. Proceedings of machine learning research, vol. 48, New York, New York, USA: PMLR; 2016, p. 1050–9, URL https://proceedings.mlr.press/v48/gal16.html.

[56] Zha Kaiwen, Cao Peng, Yang Yuzhe, Katabi Dina. Supervised contrastive regression. 2022, arXiv:2210.01189. Manuscript under review.

[57] Khosla Prannay, Teterwak Piotr, Wang Chen, Sarna Aaron, Tian Yonglong, Isola Phillip, Maschinot Aaron, Liu Ce, Krishnan Dilip. Supervised contrastive learning. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in neural information processing systems. Vol. 33, Curran Associates, Inc.; 2020, p. 18661–73, URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

[58] Chen Ting, Kornblith Simon, Norouzi Mohammad, Hinton Geoffrey. A simple framework for contrastive learning of visual representations. In: Daumé Hal, Singh Aarti, editors. Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol. 119, PMLR; 2020, p. 1597–607, URL https://proceedings.mlr.press/v119/chen20j.html.

[59] McInnes Leland, Healy John, Saul Nathaniel, Großberger Lukas. UMAP: Uniform manifold approximation and projection. J Open Source Softw 2018;3(29):861. http://dx.doi.org/10.21105/joss.00861.

[60] Buscema Massimo. Back propagation neural networks. Substance Use Misuse 1998;33(2):233–70. http://dx.doi.org/10.3109/10826089809115863, arXiv:https://doi.org/10.3109/10826089809115863. PMID: 9516725.

[61] Wang Q, Zhao B, Ma H, et al. A method for rapidly evaluating reliability and predicting remaining useful life using two-dimensional convolutional neural network with signal conversion. J Mech Sci Technol 2019;33(6):2561–71. http://dx.doi.org/10.1007/s12206-019-0504-x.

[62] Ding Pan, Liu Xiaojuan, Li Huiqin, Huang Zequan, Zhang Ke, Shao Long, Abedinia Oveis. Useful life prediction based on wavelet packet decomposition and two-dimensional convolutional neural network for lithium-ion batteries. Renew Sustain Energy Rev 2021;148:111287. http://dx.doi.org/10.1016/j.rser.2021.111287, URL https://www.sciencedirect.com/science/article/pii/S1364032121005748.

[63] Neimark Daniel, Bar Omri, Zohar Maya, Asselmann Dotan. Video transformer network. In: 2021 IEEE/CVF international conference on computer vision workshops. ICCVW, 2021, p. 3156–65. http://dx.doi.org/10.1109/ICCVW54120.2021.00355.

[64] Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina. BERT: pre-training of deep bidirectional transformers for language understanding. 2018, CoRR abs/1810.04805. URL http://arxiv.org/abs/1810.04805.

[65] Lim Bryan, Arık Sercan, Loeff Nicolas, Pfister Tomas. Temporal fusion transformers for interpretable multi-horizon time series forecasting. Int J Forecast 2021;37(4):1748–64. http://dx.doi.org/10.1016/j.ijforecast.2021.03.012, URL https://www.sciencedirect.com/science/article/pii/S0169207021000637.

[66] Xiong Ruibin, Yang Yunchang, He Di, Zheng Kai, Zheng Shuxin, Xing Chen, Zhang Huishuai, Lan Yanyan, Wang Liwei, Liu Tieyan. On layer normalization in the transformer architecture. In: Daumé Hal, Singh Aarti, editors. Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol. 119, PMLR; 2020, p. 10524–33, URL https://proceedings.mlr.press/v119/xiong20b.html.

[67] Wu Haiping, Xiao Bin, Codella Noel, Liu Mengchen, Dai Xiyang, Yuan Lu, Zhang Lei. CvT: Introducing convolutions to vision transformers. In: 2021 IEEE/CVF international conference on computer vision. ICCV, 2021, p. 22–31. http://dx.doi.org/10.1109/ICCV48922.2021.00009.

[68] Eleftheroglou N. Adaptive prognostics for remaining useful life of composite structures. 2020, URL https://repository.tudelft.nl/islandora/object/uuid:538558fb-ac9a-414d-8a59-4b523d8ff74c?collection=research.

[69] Reifsnider KL, Talug A. Analysis of fatigue damage in composite laminates. Int J Fatigue 1980;2(1):3–11. http://dx.doi.org/10.1016/0142-1123(80)90022-5, URL https://www.sciencedirect.com/science/article/pii/0142112380900225.

[70] Li Xi, Kupski Julian, Teixeira De Freitas Sofia, Benedictus Rinze, Zarouchas Dimitrios. Unfolding the early fatigue damage process for CFRP cross-ply laminates. Int J Fatigue 2020;140:105820. http://dx.doi.org/10.1016/j.ijfatigue.2020.105820, URL https://www.sciencedirect.com/science/article/pii/S0142112320303510.

[71] Kaufman Shachar, Rosset Saharon, Perlich Claudia, Stitelman Ori. Leakage in data mining: Formulation, detection, and avoidance. ACM Trans Knowl Discov Data 2012;6(4). http://dx.doi.org/10.1145/2382577.2382579.

[72] Huang Xiao Shi, Perez Felipe, Ba Jimmy, Volkovs Maksims. Improving transformer optimization through better initialization. In: Daumé Hal, Singh Aarti, editors. Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol. 119, PMLR; 2020, p. 4475–83, URL https://proceedings.mlr.press/v119/huang20f.html.

[73] Eleftheroglou Nick, Zarouchas Dimitrios, Loutas Theodoros, Alderliesten Rene, Benedictus Rinze. Structural health monitoring data fusion for in-situ life prognosis of composite structures. Reliab Eng Syst Saf 2018;178:40–54. http://dx.doi.org/10.1016/j.ress.2018.04.031, URL https://www.sciencedirect.com/science/article/pii/S0951832017306737.