Springer

Acta Mechanica Sinica

CrossMark click for updates

• Research Paper •

Acta Mech. Sin., Vol. 42, 725292 (2026)
https://doi.org/10.1007/s10409-025-25292-x

Predicting open-hole laminates failure using support vector machines with classical and quantum kernels

Giorgio Tosti Balducci[1], Boyang Chen[1*], Matthias Möller[2], Marc Gerritsma[3], and Roeland De Breuker[1]

[1] *Aerospace Structure and Materials, Delft University of Technology, Delft 2629 HS, Netherlands;*
[2] *Applied Mathematics, Delft University of Technology, Delft 2628 CD, Netherlands;*
[3] *Flow Physics and Technology, Delft University of Technology, Delft 2629 HS, Netherlands*

Received April 6, 2025; accepted June 17, 2025

Abstract: Modeling open-hole failure of composites is a complex task, consisting of a highly nonlinear response with interacting failure modes. Numerical modeling of this phenomenon has traditionally been based on the finite element method, but requires to tradeoff between high fidelity and computational cost. To mitigate this shortcoming, recent work has leveraged machine learning to predict the strength of open-hole composite specimens. Here, we also propose using data-based models to tackle open-hole composite failure from a classification point of view. More specifically, we show how to train surrogate models to learn the ultimate failure envelope of an open-hole composite plate under in-plane loading. To achieve this, we solve the classification problem via support vector machine (SVM) and test different classifiers by changing the SVM kernel function. The flexibility of kernel-based SVM also allows us to integrate the recently developed quantum kernels in our algorithm and compare them with the standard radial basis function kernel. Finally, thanks to kernel-target alignment optimization, we tune the free parameters of all kernels to best separate safe and failure-inducing loading states. The results show classification accuracies higher than 90% for RBF, especially after alignment, followed closely by the quantum kernel classifiers.

Keywords: Composites, Support vector machines, Quantum machine learning

Citation: G. Tosti Balducci, B. Chen, M. Möller, M. Gerritsma, and R. De Breuker, Predicting open-hole laminates failure using support vector machines with classical and quantum kernels, Acta Mech. Sin. **42**, 725292 (2026), https://doi.org/10.1007/s10409-025-25292-x

1. Introduction

Modern aviation industry makes wide use of composite materials, thanks to their lightweight and favorable mechanical properties. Frequently, aeronautical structural elements are not textbook flat composite panels, but tailored components with complex mechanical responses. For instance, composite panels often show cutouts in order to allow fastening or lightening the structure, or even for allowing the passage of wiring or cables. However, the presence of holes in a composite plate induces stress concentrations that can initiate damage, which can propagate into intricate failure mechanisms involving different modes.

Models for open-hole composite failure have been developed in different directions. On the one hand, semi-empirical models were proposed to predict the allowables of these structures, such as ultimate strength, and their statistical distribution with respect to hole geometry, loading conditions, stacking sequence, ply thickness, etc. Early attempts required experimental properties from testing both the unnotched and notched laminate [1], while later models removed the need to directly test the open-hole laminate [2, 3] or just required the ply properties [4]. Despite being fast to evaluate and suitable for preliminary design, semi-empirical models can make large errors when extensive delaminations propagate from the notch, as it happens with ply-scaled laminates.

Finite element (FE) simulations allow for improved modeling of open-hole laminates failure. Open-hole tension

*Corresponding author. E-mail address: b.chen-2@tudelft.nl (Boyang Chen)

(OHT) has been extensively studied numerically both for capturing the in-plane [5] and thickness size effects [6-8] on the ultimate strength and for reproducing the different failure modes and their interactions [9, 10] with increasing detail. Furthermore, FE simulations managed to quite accurately predict open-hole compression (OHC), even though they are still struggling to predict the precise kink band formation [11-13]. However, the accuracy offered by FE models generally comes at the price of high computational costs, possibly making them unfeasible when many design iterations are required.

Therefore, there is a practical need for computationally efficient yet accurate models that can simulate open-hole composite laminates. A possibility is offered by machine learning surrogates, which have been employed in composite design and optimization [14-16], constitutive law modeling, and multiscale analyses (see Ref. [17] for a comprehensive review) and damage characterisation [18, 19]. Concerning open-hole composite failure, Furtado et al. [20] proposed a methodology to define allowables using four different machine learning models. Their methodology was applied to open-hole tensile strength prediction for different dimensions, layups, and material properties. While their methods are demonstrated on data generated analytically [4], the authors suggest using high-fidelity finite element analyses for training, potentially providing accurate data-based models.

Similarly, in this work, we propose a machine learning surrogate for open-hole composites, which is accurate and efficient in inference. Different from Ref. [20], the approach we suggest is not to have a fast allowables generator, but a classifier for the ultimate failure of open-hole composite laminates. More precisely, our trained model takes a loading state as input, such as the far field homogenized plane strain components, and returns a binary value (± 1) as output, depending on whether the load applied is lower or higher than the notched laminate strength. In this sense, the surrogate acts as a data-based generalized failure criterion that predicts at the structural component level, rather than at the material level. The training data are obtained from non-linear FEA with multi-directional strain loadings on the open-hole panel. As opposed to a uniform grid, Sobol sequence, or Latin Hypercube sampling over the full strain space, a radial sampling strategy is employed to essentially sample only the directional vectors of the strain space. Simulations are run until total failure of the open-hole specimen is reached, and the incremental solutions are stored as intermediate stress-strain data. Therefore, sampling is more efficient as a single sampling generates a lot of stress-strain data.

This paper also aims to compare classical and quantum computation for a classification problem in composite mechanics. To do this, we train the machine learning surrogate using kernel-based support vector machines (SVMs) [21], where the kernel function can be computed both in classical and quantum logic. As it will be clear in the next sections, quantum computation offers a way to encode information into exponentially large Hilbert spaces and to define an inner product in these spaces, effectively generating a kernel. This allows for exploring the generalization potential of quantum machine learning, while leaving the SVM optimization to well-established classical quadratic optimization algorithms.

The rest of the paper is structured as follows. Section 2 describes the machine learning problem by defining the input, the data sampling strategy, and the labeling criterion. Section 3 briefly introduces the SVM dual problem, the radial basis functions (RBF) kernels, and the quantum kernels. More details about these methods are available in the appendices following the main body of the manuscript. Finally, Sect. 4 presents the classification results for all kernels and Sect. 5 outlines conclusions and future work.

All data and code used in this work are made publicly available [1) 2)].

2. Machine learning problem

Our method was applied to predict failure of an open-hole composite specimen similar in geometry and material properties to the one experimentally tested in Ref. [22]. The specimen was modeled and meshed with the Abaqus finite element code [23], and it was loaded with different combinations of axial and shear strains and constrained with periodic boundary conditions. All the details of the specimen properties and of the finite element analyses are left to Appendix A.

The input of our surrogate models are homogenized far field strains $\boldsymbol{\varepsilon} = [\varepsilon_{11}, \varepsilon_{22}, \gamma_{12}]^{\mathrm{T}}$, which derive from enforcing periodic boundary conditions on opposite faces of the plate. The displacements of the left/right and top/bottom faces, respectively, can be linked through some reference degrees of freedom

$$\begin{aligned} U_1 &= u_1^{\mathrm{R}} - u_1^{\mathrm{L}}, \quad U_2 = u_2^{\mathrm{T}} - u_2^{\mathrm{B}}, \\ U_3 &= u_2^{\mathrm{R}} - u_2^{\mathrm{L}}, \quad U_4 = u_1^{\mathrm{T}} - u_1^{\mathrm{B}}, \end{aligned} \tag{1}$$

where directions 1 and 2 are the horizontal and vertical directions in Fig. 1. The homogenized strains are then obtained

1) Boyang Chen and Giorgio Tosti Balducci. Nonlinear responses of metals and composites. https://doi.org/10.5281/zenodo.7409612.

2) Giorgio Tosti Balducci. oh-comp-kernels: A Python code for the prediction of the open-hole tensile strength of composite laminates using kernel methods. https://github.com/debrevitatevitae/oh-comp-kernels.

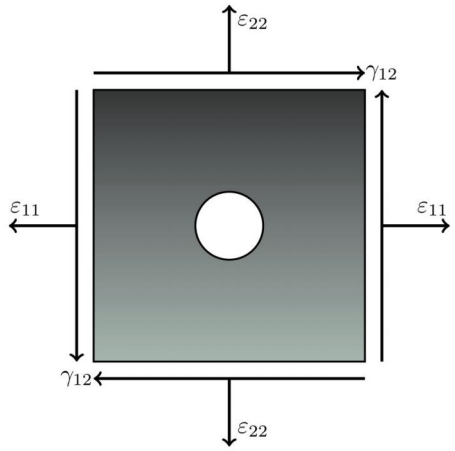

Figure 1 Geometry of the open-hole composite plate and homogeneous strain loads components.

as

$$\varepsilon_{11} = \frac{U_1}{D_1},\ \varepsilon_{22} = \frac{U_2}{D_2},\ \gamma_{12} = \frac{U_3}{D_1} + \frac{U_4}{D_2}, \tag{2}$$

where D_1 and D_2 are the planar dimensions of the plate.

As mentioned, the input space was sampled through nonlinear incremental-iterative finite element analyses. Figure 2 illustrates the sampling strategy used in this work in the simplified case of two-dimensional input. We refer to this technique as radial sampling, due to the fact that the design of experiments (DoE) does not directly affect all the points in this input space, but only the ones on the boundary. On the other hand, all the intermediate points are generated internally by the FE solver, and they correspond to the homogenized strain values at every time increment. The user maintains control of the inner sample values by the choice of initial, minimum, and maximum time steps. For this work, we chose the sampling space to be the hypercube $\left[-10^{-2},\ 10^{-2}\right]^{\otimes 3}$ in $\mathbb{R}^3$, meaning that all three components of the applied strains vector have the same bounds.

Each strain sample was assigned a label based on an ultimate failure criterion. In particular, we defined failure by the loss of stiffness of the laminate for given a user-defined threshold.

From the results of the FE analyses with periodic boundary conditions, one obtains the reaction forces F_1, F_2, F_3, and F_4 conjugate to the degrees of freedom in Eq. (1). These provide the homogenized stresses, which can then be derived via the Hill-Mandel principle of energy balance as

$$\sigma_{11} = \frac{F_1}{tD_2},\ \sigma_{22} = \frac{F_2}{tD_1},\ \sigma_{12} = \frac{F_3U_3 + F_4U_4}{\gamma_{12}tD_1D_2}, \tag{3}$$

where t is the thickness of the plate.

The laminate stiffness in the two axial directions and in shear can thus be defined at every timestep t as

$$E_1^{(t)} = \frac{\sigma_{11}^{(t)}}{\varepsilon_{11}^{(t)}},\ E_2^{(t)} = \frac{\sigma_{22}^{(t)}}{\varepsilon_{22}^{(t)}},\ G_{12}^{(t)} = \frac{\sigma_{12}^{(t)}}{\varepsilon_{12}^{(t)}}. \tag{4}$$

The stiffness degradation d_S is defined as the minimum ratio between the instantaneous stiffness and the corresponding stiffness measure in the linear elastic region,

$$d_S^{(t)} = \min\left\{\frac{E_1^{(t)}}{E_1^{(0)}}, \frac{E_2^{(t)}}{E_2^{(0)}}, \frac{G_{12}^{(t)}}{G_{12}^{(0)}}\right\}. \tag{5}$$

Therefore, given M the total number of samples, every sample $\varepsilon^{(m)}$ $(m = 1, 2, \ldots, M)$ is assinged a label $y^{(m)} = -1$ if $d^{(m)} < \bar{d}_S$ and $y^{(m)} = +1$ otherwise.

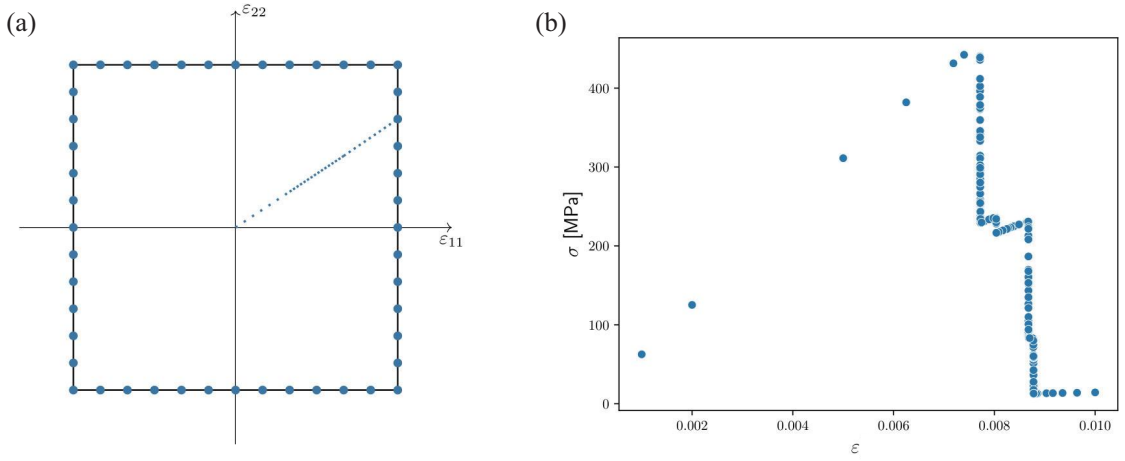

Figure 2 (a) Radial strategy for sampling the far-field strain space. The bigger dots on the edge of the $(\varepsilon_{11}, \varepsilon_{22})$ volume represent the final applied load in different nonlinear FE incremental-iterative analyses. The arrow represents a loading path with the load increments unevenly distributed. (b) Load-displacement curve corresponding to the loading path in (a).

3. Methodology

As already mentioned, we solve the ultimate failure binary classification problem using the SVM algorithm [21]. This consists of the following quadratic optimization problem in dual form:

$$\begin{aligned} &\max_{\alpha} \sum_{m=1}^{M} \alpha^{(m)} - \frac{1}{2} \sum_{m,m'=1}^{M} y^{(m)} y^{(m')} \alpha^{(m)} \alpha^{(m')} \kappa\left(\boldsymbol{x}^{(m)} \boldsymbol{x}^{(m')}\right), \\ &\text{s.t. } 0 \le \alpha^{(m)} \le C, \quad m = 1, 2, \ldots, M, \\ &\sum_{m=1}^{M} \alpha^{(m)} y^{(m)} = 0, \end{aligned} \tag{6}$$

where $\boldsymbol{x}^{(m)} = \boldsymbol{\varepsilon}^{(m)}$ and $y^{(m)} = \pm 1$ are the labels, respectively non-failed and failed; $\alpha^{(m)}$ are the Lagrange multipliers; C is the slack penalty; the kernel function $\kappa(\cdot, \cdot)$ is a similarity metric between two samples in a higher-dimensional feature space. More details on the SVM algorithm are left to Appendix B.

The performance of the dual SVM depends on the choice of its hyperparameters, namely the kernel function κ and slack penalty C. To restrict the search space, the kernel function is generally parametrized via one or more parameters $\boldsymbol{\theta}$, and the standard practice is to perform a grid-search cross-validation procedure in the $(\boldsymbol{\theta}, C)$ space. In this work, we use instead a mixed procedure, where the kernel function is determined by optimizing the kernel-target alignment (KTA) [24] and the slack penalty is found by grid search cross-validation. The overall methodology is illustrated in Fig. 3, where we refer to the two steps as kernel training and SVM selection. The "test set scoring" step of the model is meant to assess the generalizability of the model on unseen data from the training. Once the SVM has been fully determined, it can be trained by solving Eq. (6) and its learning ability can be measured as the accuracy on unseen test data, for different training dataset sizes.

We compare one classical and two quantum kernels. The classical kernel is the radial basis function (RBF) kernel, defined as

$$\kappa_{\mathrm{RBF}}\left(\boldsymbol{x}^{(m)}, \boldsymbol{x}^{(m')}\right) = \exp\left(-\gamma \left\|\boldsymbol{x}^{(m)} - \boldsymbol{x}^{(m')}\right\|^2\right). \tag{7}$$

RBF is a powerful kernel that corresponds to a feature map in an infinite-dimensional feature space [25]. It induces a Gaussian similarity function, whose width is controlled by the hyperparameter γ.

On the other hand, the quantum kernel is defined via a quantum embedding, which is constructed via data-dependent unitary transformations $U(\boldsymbol{x})$ that prepare the quantum state

$$\left|\psi(\boldsymbol{x})\right\rangle = U(\boldsymbol{x}) \left|0\right\rangle. \tag{8}$$

Given two samples $\boldsymbol{x}^{(m)}$ and $\boldsymbol{x}^{(m')}$, the quantum kernel is simply the inner product

$$\kappa_{\mathrm{Q}}\left(\boldsymbol{x}^{(m)}, \boldsymbol{x}^{(m')}\right) = \left|\left\langle \psi\left(\boldsymbol{x}^{(m)}\right), \psi\left(\boldsymbol{x}^{(m')}\right)\right\rangle\right|^2. \tag{9}$$

Figure 4 shows the generic quantum embedding and the two specific ones used in this work, which are the hardware-

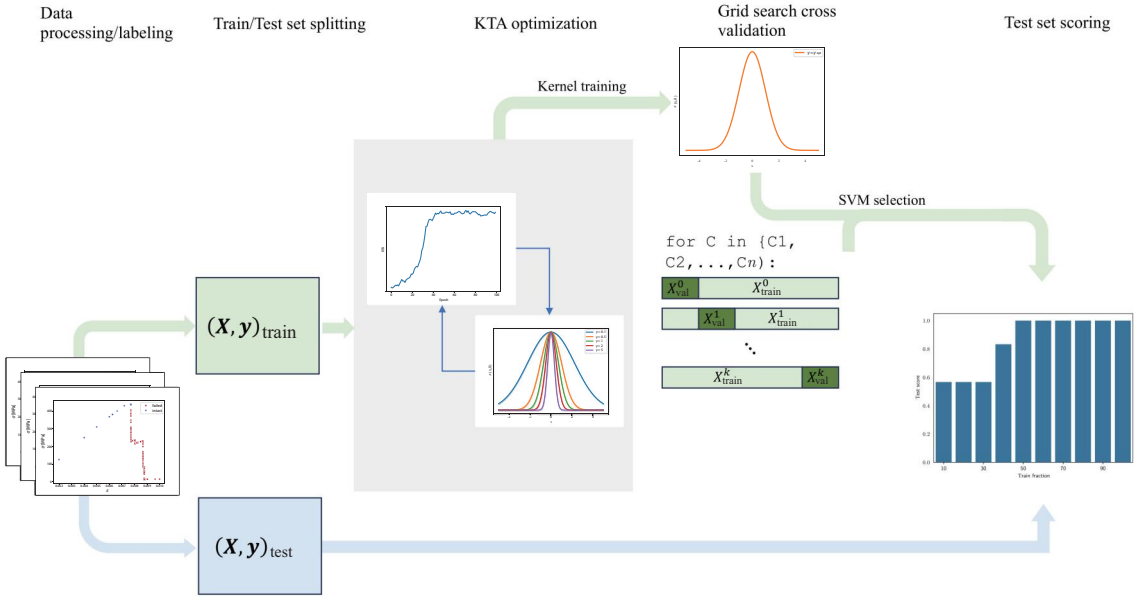

Figure 3 Methodology used in this work. The dataset is generated by nonlinear finite element analyses, then labeled and split into training and testing sets. The training set is used first to train the kernel, by optimizing the KTA, then in a grid search cross-validation to find the best slack penalty C of the SVM. With all the hyperparameters fixed, the SVM is trained for increasing dataset sizes, and the classification accuracy is evaluated on the testing set.

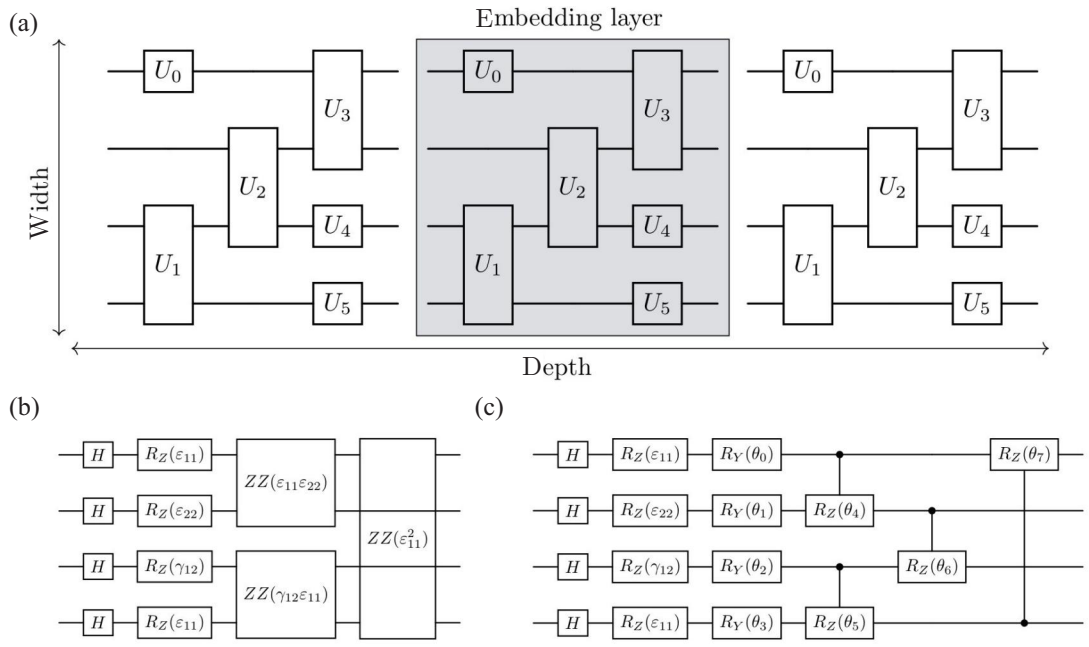

Figure 4 Quantum embeddings. (a) Generic quantum embedding made of one- and two-qubit gates. The width of the embedding is the number of qubits, while the depth is the number of layers, which is a minimal block of gates. (b) IQP quantum embedding layer, which is parameter-free, but encodes products of features as ZZ interactions. (c) HE2 quantum embedding layer, parametrized by $[\theta_0, \theta_1, \ldots, \theta_{P-1}]$.

efficient embedding (HE2) [26] and the instantaneous quantum polynomial (IQP) [27] one. To have more expressive feature mapping, either the width or the depth of the quantum embedding can be increased. The first one is the number of qubits, which can be even higher than the number of features in the dataset, by cyclically re-encoding the features to generate a highly nonlinear and potentially better separable feature space. Meanwhile, the embedding's depth can be increased by repeating a base data-encoding block, such as IQP and HE2. Even in this case, re-encoding of the features may lead to a higher expressivity of the overall feature map [28]. For a short summary of relevant quantum computing concepts, we refer the reader to Appendix C.

4. Results

We tested our machine learning models on a dataset of 1960 labelled strain vectors $\boldsymbol{\varepsilon}^{(m)}$, which we obtained by uniformly sampling the homogeneous strain/stress pairs from the FE simulations of the open-hole composite specimen. The test set size is equal to 20% of the size of the total dataset (1960 samples). Therefore, we isolated 392 test samples. The input homogeneous strains in both normal and shear directions were varied between 10^{-4} and 10^{4} microstrains, and a stiffness degradation threshold of 0.9 was used to discriminate non-failed and failed loading states. 0.9 was deemed sufficiently large such that the residual stiffness of the structure can be neglected. In practice, this number would depend on case requirements.

Both classical- and quantum-kernel SVMs were implemented using different Python libraries. We used PyTorch for training the RBF kernel and PennyLane for the quantum kernels. These libraries implement automatic differentiation (AD), which allows for optimizing the KTA with gradient-based methods. We also used JAX together with PennyLane to just-in-time compile the quantum kernel functions. Concerning the classification problem, we employed the SVM and grid-search cross-validation routines available from the Scikit-Learn Python package.

The KTA of both RBF and quantum kernels was maximized using stochastic gradient descent and Adam parameters update [29]. Figure 5 shows the kernel alignment training of the RBF kernel. Figure 6 presents instead the KTAs before and after training for 12 different quantum kernels with HE2 embedding. It can be seen that increasing the width and depth of these kernels generally improves their KTA. A higher number of qubits means that the strain features are mapped in a higher-dimensional space, which can favor the separability of the classes. On the other hand, increasing the depth benefits the kernel alignment, since it results in more expressive feature maps. Also, every additional layer of the HE2 embedding doubles the number of free parameters, explaining why optimization of deeper kernels mostly leads to higher gains in KTA. However, the advantage of increasing these quantum encoding resources does not scale uniformly. Already with 6 qubits and 3 HE2 layers, the optimization

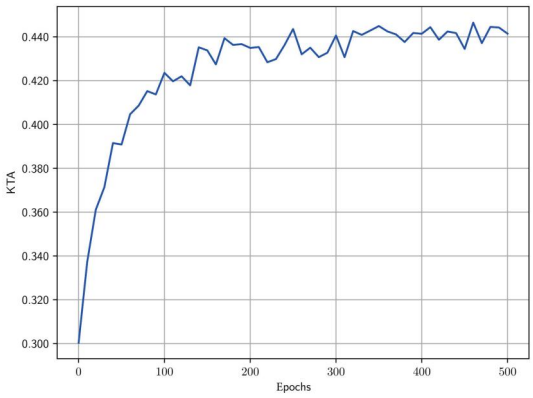

Figure 5 Training history of the RBF kernel's KTA.

only modestly improves the KTA, likely due to the vanishing KTA gradients [30].

To find the hyperparameter C that guarantees the highest off-training accuracy of the SVM algorithm, we used grid-search cross-validation for the kernels considered. The validation accuracy values are reported in Fig. 7 for multiple values of C and γ. We observe that kernels with $\gamma \leq 10$ achieve the highest scores, with the highest-KTA γ scoring first for the whole range of C values. Furthermore, the accuracy of the maximally aligned RBF kernel increases monotonically with C, which suggests the usefulness of maximizing the KTA, but also that the class boundary in this feature space is densely populated and still requires a tight margin.

The same analysis was performed for all the quantum kernels considered, where we wanted to take into account the effect on accuracy of different embeddings and of maximizing the KTA. The results are reported in Fig. 8, which shows accuracies roughly between 67% and 87% for all embeddings with different values of C. Except for the IQP case, increasing C leads to higher accuracies, hinting at the need for a tight bound when mapping with these embeddings, similar to the RBF kernel. Unfortunately, for high values of C, the optimization of the dual SVM failed to converge for some of the quantum kernels, likely due to numerical ill-conditioning. This presumably prevented the quantum kernel classifiers from even better separating failed instances, as suggested by the monotonic increasing validation accuracies with C, at least for the HE2 kernels. Furthermore, Fig. 8 shows that the scores improve when more embedding resources (number of qubits and layers) are added, especially in the case of KTA-optimized HE2 kernel. The numerical instability of the dual SVM with quantum kernels could be due to a very slow convergence of the optimizer when the slack penalty is large. The SVM backend in our library is libsvm, which solves the quadratic problem with the sequential minimal optimization (SMO) algorithm. For high values of slack penalty C, this solver is known to slow down around the optimal solution [31, 32]. Although we tried to relax the solver's tolerance to meet the KKT conditions, our quantum kernel SVMs still did not converge in a meaningful time. A potential remedy for this shortcoming would be to change the method of solving the SVM to the active set method [31], which has shown that the high-penalty instances could recover the convergence rate of the low-penalty ones [32].

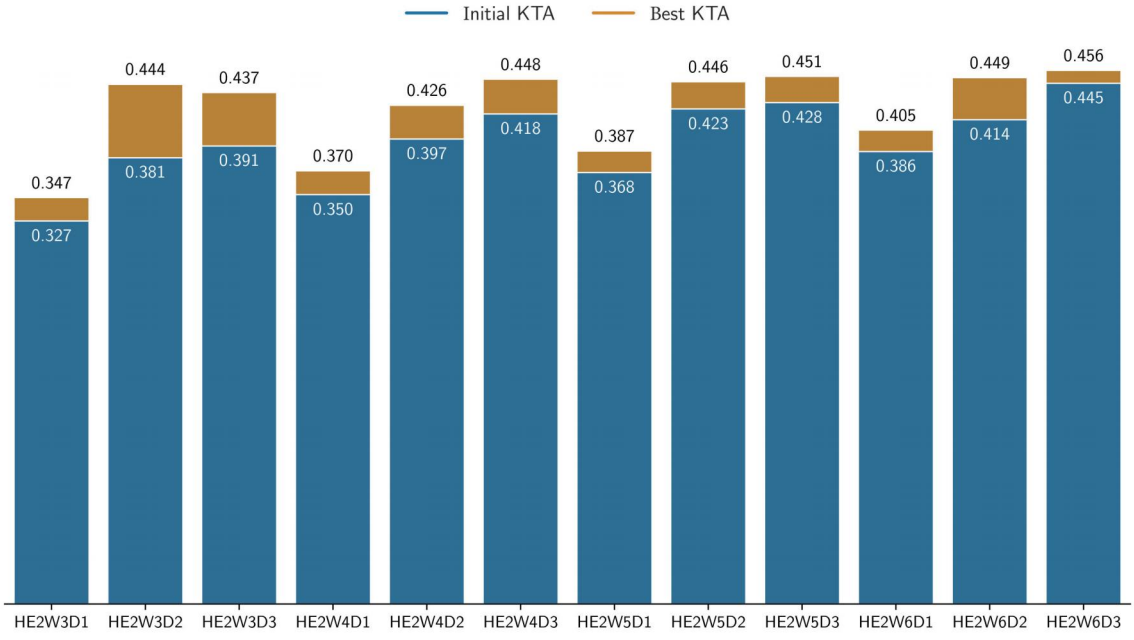

Figure 6 KTA of quantum kernels before and after training for 12 different quantum embeddings. The basic layer for all embeddings is the HE2, which has trainable parameters. The tag WXDY indicates the width and depth of the embedding.

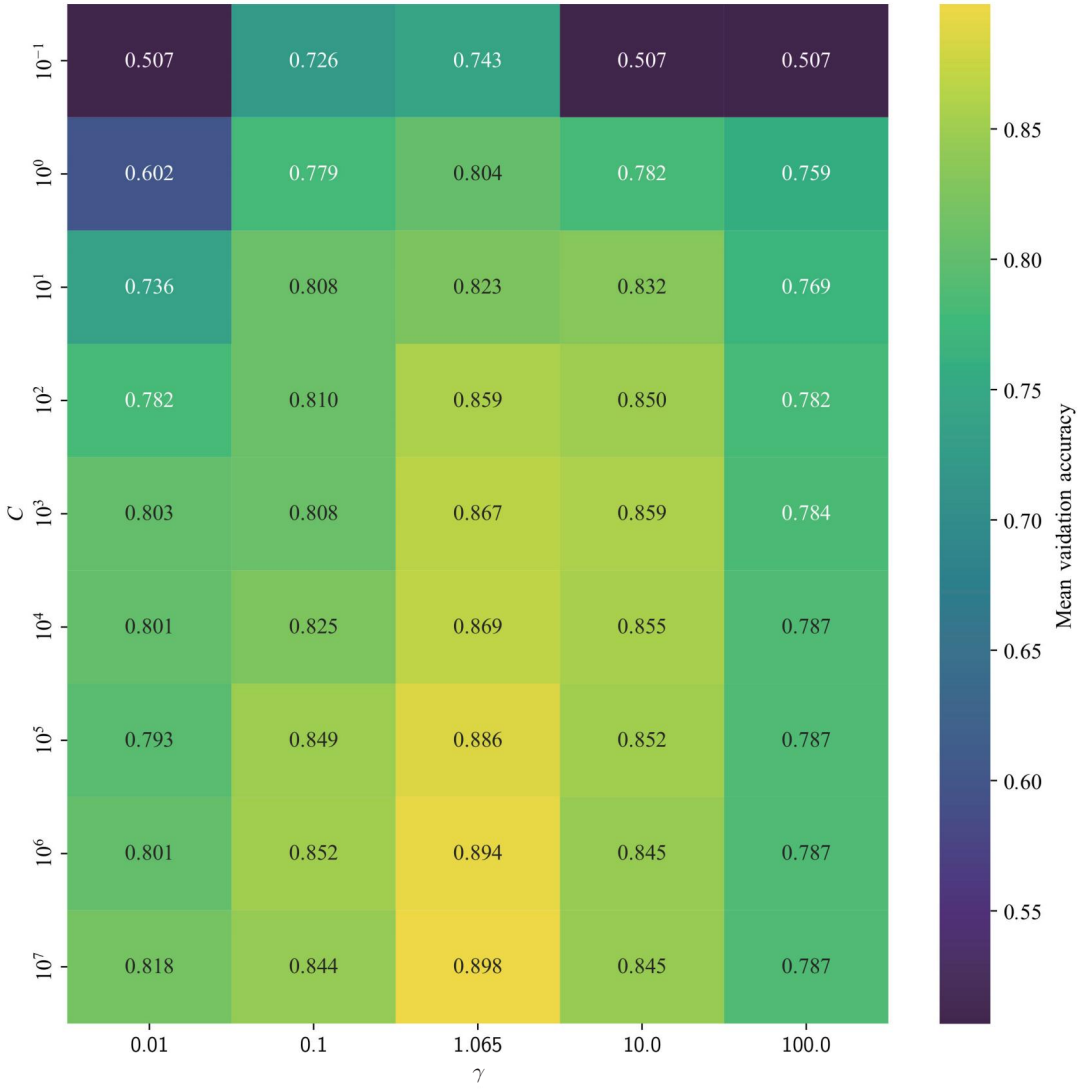

Figure 7 Grid-search cross validation for the RBF kernel. $\gamma = 1.065$ corresponds to the kernel with the highest KTA.

Classical and quantum kernels are finally compared in Fig. 9, which shows how 5 different models classify a test set of strain loading data when fitted on progressively larger training sets. A similar comparison on additional classification metrics can be found in Appendix D. The RBF kernel achieves 80% accuracy with just 10% of the total training set size, and with $C = 10^7$ it reaches over 90% with just half the training points. In comparison, all quantum kernel classifiers are at least 5% less accurate than the best RBF-kernel SVM. However, especially for HE2 embeddings, the scores are similar to the $C = 10^4$ RBF case, suggesting that RBF and HE2 kernels separate the non-failed and failed classes to a similar extent. Changing the embedding from HE2 to IQP, there is a drop in accuracy for small training set sizes, while the performance is similar when more than half the training set is used. On the other hand, the effect of training the kernel is less visible at this stage, reflecting the fact that the accuracies obtained during grid-search cross-validation are alike for untrained and trained HE2.

The efforts beyond the coding and the preparation of data would mostly be in the computational modelling for the data generator. In our experience, data generation is where most computational time is spent. To cover the input space, the method needs many nonlinear incremental analyses for each strain-loading applied. Depending on the complexity of the model, the nonlinearities of the problem, and the number of samples, this can take from hours to days. In comparison, training and prediction are faster by orders of magnitude. The expensive step in training is the optimization of the KTA. Nevertheless, a common gradient-based optimizer, like Adam, reaches good alignment in minutes at most. Resolution of the SVM quadratic problem can happen in seconds, in general. Concerning predictions, the evaluation of the trained SVM is basically instantaneous, especially when

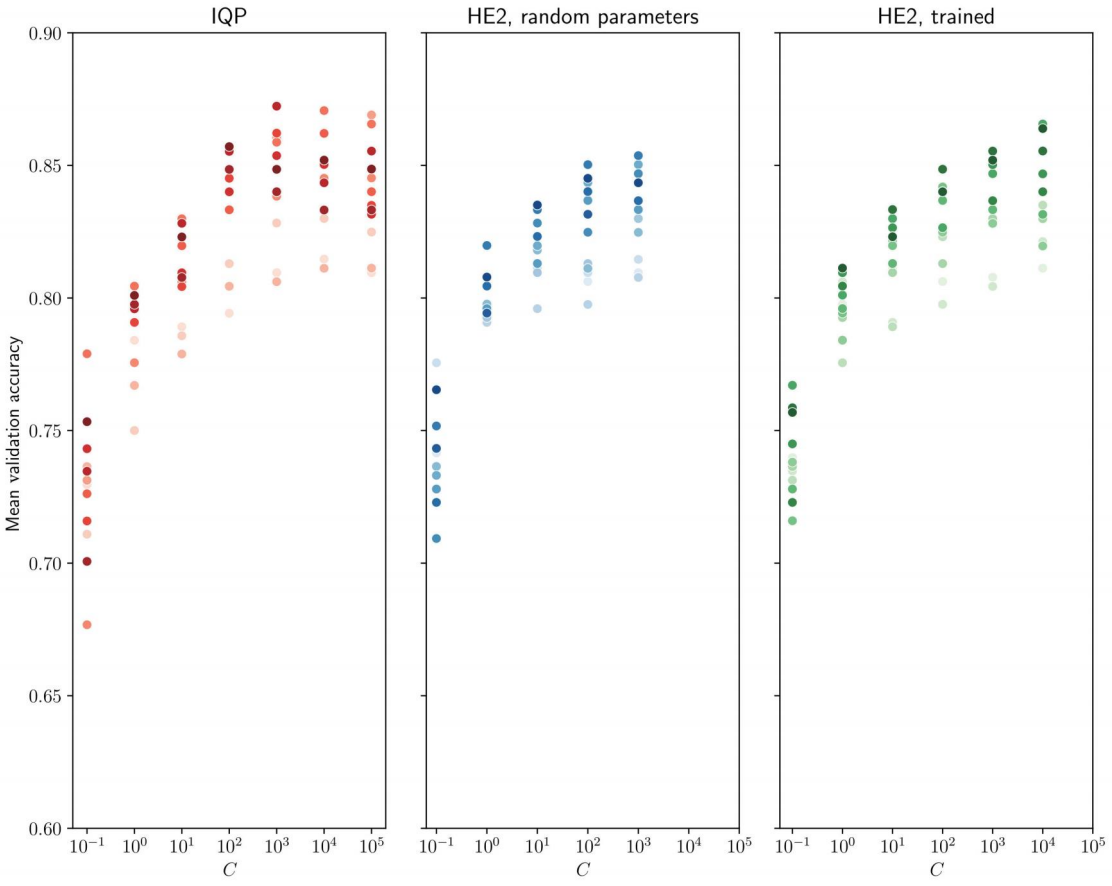

Figure 8 Summary of the grid-search cross-validation results for different quantum kernels. The three figures correspond respectively to the IQP embedding, the HE2 kernel with untrained parameters, and HE2 with trained parameters. Different shades of the same color correspond to different depths and widths. From lighter to darker, the points correspond to embeddings of increasing widths and of increasing depth per fixed width.

a few support vectors are needed. A note about querying quantum hardware is that it should happen in large batches; otherwise, the latency of communicating classical information and possibly the queue time might become bottlenecks.

5. Conclusions

In this paper, we proposed a methodology to build a binary classifier from finite element analysis data for the particular case of an open-hole composite specimen. We studied the case of in-plane strain loading of the specimen, where the objective is to correctly label strain combinations that lead to ultimate failure.

From a design of experiment point of view, we demonstrated a radial sampling strategy technique, where the choice of which simulations to make to cover the input space takes into account the incremental-iterative nature of the nonlinear FE method. We then proposed a labelling criterion of homogenized strain-stress pairs based on residual in-plane stiffness.

For classification of the labelled data, we used the kernel-based SVMs, which also allowed us to compare the performance of the recently proposed quantum kernels against the more traditional RBF. Furthermore, we employed kernel-target alignment to improve class separability of both RBF and the HE2 embedding kernel. The novelty of using quantum kernels in this failure identification exercise is to test the expressivity of quantum feature mapping in a practical scenario and to compare it with classical (implicit) mappings, such as RBFs. This is relevant because it is believed that higher expressivity is one of the potential advantages of quantum machine learning over classical. Furthermore, relying on kernels, rather than heuristics such as (quantum) neural networks, allows us to isolate the effects of expressivity in models' comparison.

For all the kernels examined, the corresponding SVMs separate non-failed and failed loading states with good accuracy. The RBF-based model classifies more accurately than its quantum counterparts, although this likely happens due to numerical ill-conditioning in the current quantum SVM implementation. These numerical issues can likely be fixed by studying the dual SVM problem for the problematic instances, which will be the subject of future work.

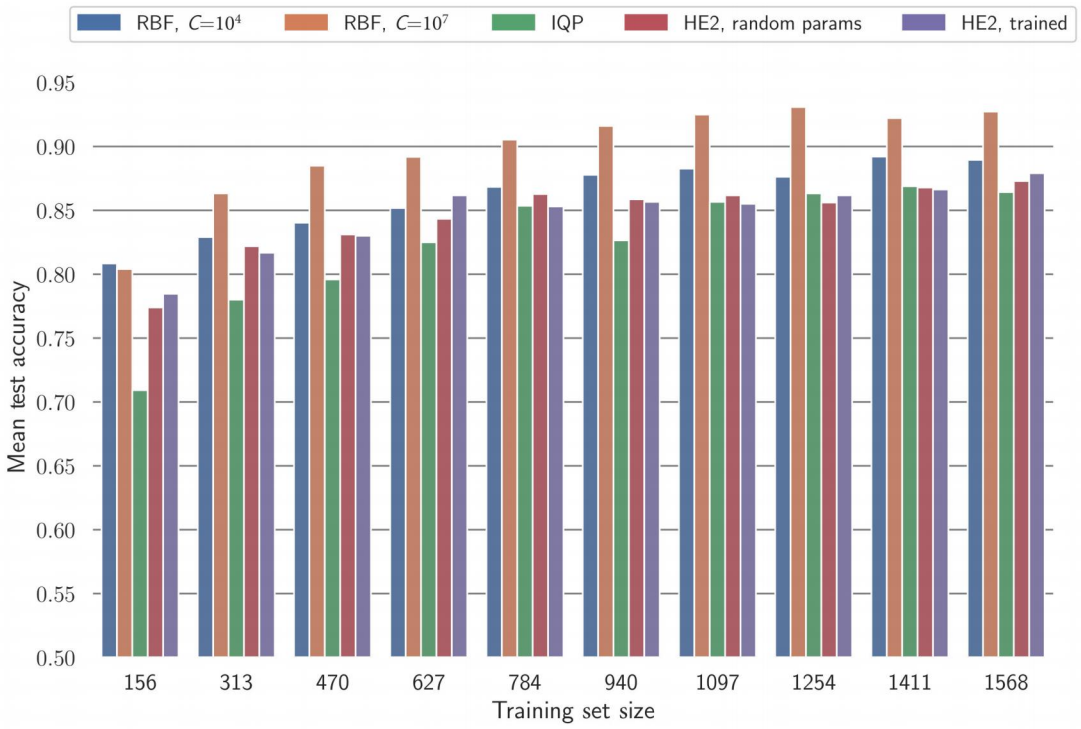

Figure 9 Test data classification accuracy of classical and quantum kernels for increasing training set size. The RBF kernel-SVMs were trained with both $C = 10^7$ and $C = 10^4$, as the latter is the highest value of C for which the quantum-kernel SVMs could still be fitted. Quantum-kernel classifiers were trained instead with the embedding architecture and C value, ensuring the highest accuracy during grid-search cross validation.

Regarding kernel alignment, optimizing the KTA is shown to be powerful for RBF, since the SVM for the trained kernel outperforms the other RBF-based models in terms of accuracy. Aligning quantum kernels for this dataset also helps them to better separate the two classes, but for simple architectures, the improvement is moderate, while more complex embeddings only reach the scores of the simpler ones after they have been aligned. Furthermore, one should remember that optimizing quantum kernels is almost always more computationally involved than for RBF, as the former can have highly parametrized embeddings, while RBF is completely defined by the single parameter γ.

Extensions of this work can go in many directions. From the point of view of the problem, it would be interesting to increase the number of degrees of freedom by allowing the notch radius or the lamination sequence to also change. The latter could be written in terms of lamination parameters [33] to have a continuous representation. The FEA model here serves as a relatively simple, proof-of-concept type of data generator. It lacks features such as delamination, discrete matrix cracks, and (local) buckling in its model. Hence, its predictions were not expected to be of high fidelity so as to be validated with experimental data. The overall methodology of training (quantum) kernel-based SVM as a data-driven failure envelope for open-hole laminate, however, remains valid despite the simplicity of the chosen FE model. In future work, we intend to replace the simple FE model with more sophisticated, discrete-crack models based on XFEM or FNM, where the predictions would be deemed as high fidelity, which would be validated against experimental data.

In terms of algorithms, both classical and quantum kernels can be explored further. RBF is the most popular choice for classical kernels, but certainly not the only one. Due to Mercer's condition, any function that defines a positive semi-definite kernel matrix is a valid kernel function [25]. Obviously, the design space is vast, but automated procedures help reduce the search, for instance, by exploring combinations of only a fixed set of standard kernel functions.

On the other hand, the freedom of designing and parametrizing quantum embedding circuits also makes the choice of a quantum kernel nontrivial. Within the limits of classical simulation of quantum circuits, one could experiment with an increasing number of qubits or different layering strategies, for instance, the one proposed in Ref. [34] for the task of satellite image classification. From an optimization point of view, a recent technique has been proposed to maximize the quantum kernel alignment KTA and solve the SVM in a single optimization loop [35], which would greatly reduce the computational cost. Nevertheless, to truly understand the potential competitiveness of quantum kernels, it is probably most important to remove layers of simulation and study the effects of statistical and hardware noise

on SVM convergence and accuracy. In a realistic scenario, where quantum kernel values are sampled from a quantum computer, the disadvantage for quantum-SVM is the latency in classical-quantum communication of data. If a quantum kernel is able to generalize on a few instances better than a classical kernel, there might be an advantage. Furthermore, because circuits of a few qubits can be emulated classically, a candidate problem for advantage is also one for which the accuracy of the solution scales with the number of qubits.

Appendix A. Open-hole specimen features and finite element model details

A1 Geometry and material properties

The plate's hole has a 6 mm diameter, and the in-plane dimensions are both 5 times the hole diameter. The ply material is IM7/8552 prepreg (carbon fibres and epoxy matrix), and each ply has $t = 0.125$ mm thickness. We considered the lamination sequence $[45/90/-45/0]_S$ for a total of 8 plies and 1 mm laminate thickness.

A2 Details of the FE models

All finite element models were done using the Abaqus finite element code [23], and Python scripting was used to automatically generate different FE models for each of the strain loading combinations [3) 4)].

The meshed part is illustrated in Fig. A1, which shows that a radial mesh was obtained by seeding the hole edge 4 times as much as the outer edges. Since no delaminations were expected due to the absence of ply blocks, the elements were chosen to be S4 shell elements of the Abaqus standard element library [36], whose in-plane and bending behaviour are described by the classical lamination theory (CLT), once the stacking sequence and ply thicknesses are specified.

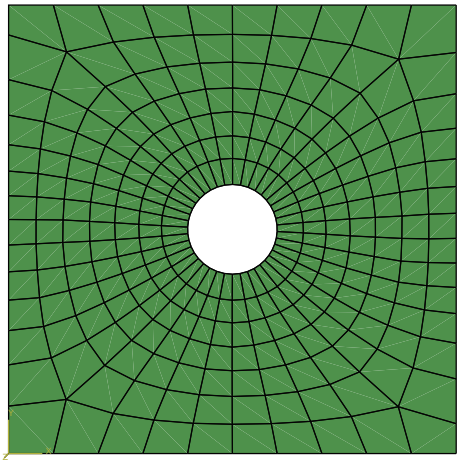

Figure A1 Finite element mesh of the composite plate.

Damage initiation was modeled with the Hashin criterion, while damage evolution was represented in a smeared crack fashion. For this purpose, the cohesive law available in Abaqus [36] was employed to model the stiffness degradation due to matrix and fiber tensile and compressive failure.

Appendix B. Support vector machines, kernel methods and KTA

B1 Primal SVM

The SVM is the linear decision model

$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b, \tag{B1}$$

which assigns labels through the sign function

$$\mathrm{sgn}(y) = \mathrm{sgn}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b\right). \tag{B2}$$

In Eqs. (B1) and (B2), $\boldsymbol{w}$ is the vector normal to the decision hyperplane, and b is the intercept.

The optimal hyperplane is found by maximizing the geometric margin of the dataset, which can be proved to be

$$\gamma^* = \frac{1}{\|\boldsymbol{w}\|}. \tag{B3}$$

By minimizing the squared norm $\|\boldsymbol{w}\|^2$ one obtaines the primal optimization problem of the SVM,

$$\begin{aligned} &\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2, \\ &\text{s.t.} \quad y^{(m)}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(m)} + b\right) \geq 1 \quad m = 1, 2, \ldots, M, \end{aligned} \tag{B4}$$

where m identifies the sample, and M is the total number of training samples.

Equation (B4) enforces exact separability, which can lead to overfitting. A way to improve generalization is the so-called soft margin SVM, which modifies Eq. (B4) by introducing the constraints slack variables $\xi^{(m)}$ and the penalty constant C,

$$\begin{aligned} &\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{m=1}^{M}\xi^{(m)}, \\ &\text{s.t.} \quad y^{(m)}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varepsilon}^{(m)} + b\right) \geq 1 - \xi^{(m)} \quad m = 1, 2, \ldots, M, \\ &\qquad \xi^{(m)} \geq 0. \end{aligned} \tag{B5}$$

3) Tom Gulikers. Computational framework to implement an artificial neural network-based constitutive model in Abaqus for mesh coarsening. https://github.com/tgulikers/ABAQUS_ANN_constitutive_model.

4) Boyang Chen. Training ANNs with FEM data on Open-hole Composite plate—for summer school 2022 in Delft. https://github.com/BoyangChenFEM/Summer2022.

B2 Dual SVM and kernels

By introducing the Lagrange multipliers $\alpha^{(m)}$ and $\beta^{(m)}$, one can write the Lagrangian of the SVM optimization problem,

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) =& \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{w}\rangle + C\sum_{m=1}^{M}\xi^{(m)} - \sum_{m=1}^{M}\alpha^{(m)}\left[y^{(m)}\right. \\ &\left.\cdot\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(m)} + b\right) - 1 + \xi^{(m)}\right] - \sum_{m=1}^{M}\beta^{(m)}\xi^{(m)}.\end{aligned} \tag{B6}$$

The dual soft-margin SVM is obtained by setting all the derivatives of the Lagrangian in Eq. (B6) equal to zero,

$$\begin{aligned}&\max_{\alpha}\ \sum_{m=1}^{M}\alpha^{(m)} - \frac{1}{2}\sum_{m,m'=1}^{M} y^{(m)}y^{(m')}\alpha^{(m)}\alpha^{(m')}\left\langle \boldsymbol{x}^{(m)}, \boldsymbol{x}^{(m')}\right\rangle \\ &\text{s.t. } 0 \le \alpha^{(m)} + \beta^{(m)} \le C, \quad m = 1, 2, \ldots, M, \\ &\qquad \sum_{m=1}^{M}\alpha^{(m)}y^{(m)} = 0.\end{aligned} \tag{B7}$$

Equation (B7) is still a linear model in the original feature space. However, by introducing a feature map

$$\phi : \boldsymbol{x} \longrightarrow \phi(\boldsymbol{x}), \tag{B8}$$

we can map the features nonlinearly and potentially to a manifold where they are more easily separable. Furthermore, replacing $\boldsymbol{x}$ with $\phi(\boldsymbol{x})$ in Eq. (B7), we see that the mapped features only appear in the inner product

$$\kappa\left(\boldsymbol{x}^{(m)}, \boldsymbol{x}^{(m')}\right) = \left\langle \phi\left(\boldsymbol{x}^{(m)}\right), \phi\left(\boldsymbol{x}^{(m')}\right)\right\rangle, \tag{B9}$$

which is known as the kernel of the feature map. The advantage of having only inner product of features (kernel trick) is the possibility of classifying in nonlinear feature spaces without having to compute the feature map explicitly.

The kernels mostly used in machine learning are the polynomial, Gaussian, and sigmoid kernels

$$\kappa(\boldsymbol{x}, \boldsymbol{x'}) = \begin{cases}(\gamma\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x'} + c_0)^d, & \text{(polynomial)},\\ \exp\left(-\gamma\|\boldsymbol{x} - \boldsymbol{x'}\|^2\right), & \text{(Gaussian)},\\ \tanh\left(\gamma\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x'} + c_0\right), & \text{(sigmoid)}.\end{cases} \tag{B10}$$

B3 Kernel-target alignment

The alignment between two kernels is defined as

$$\begin{aligned}A\left(\boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)}\right) &= \frac{\left\langle \boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)}\right\rangle_F}{\sqrt{\left\langle \boldsymbol{K}^{(1)}, \boldsymbol{K}^{(1)}\right\rangle_F \left\langle \boldsymbol{K}^{(2)}, \boldsymbol{K}^{(2)}\right\rangle_F}} \\ &= \frac{\left\langle \boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)}\right\rangle_F}{\left\|\boldsymbol{K}^{(1)}\right\|_F \left\|\boldsymbol{K}^{(2)}\right\|_F},\end{aligned} \tag{B11}$$

where $\boldsymbol{K}$ is the kernel matrix, obtained by taking the kernel of all pairs of features, and

$$\left\langle \boldsymbol{K}^{(1)}, \boldsymbol{K}^{(2)}\right\rangle_F = \mathrm{tr}\left(\boldsymbol{K}^{(1)\mathrm{T}}\boldsymbol{K}^{(2)}\right). \tag{B12}$$

The alignment between two kernels is always less than or equal to 1, where 1 corresponds to perfect alignment.

Assume a kernel $\kappa_{\boldsymbol{\theta}}$, parametrized by $\boldsymbol{\theta}$ and define the target kernel matrix as

$$\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}. \tag{B13}$$

The KTA of $\kappa_{\boldsymbol{\theta}}$ is the alignment between the chosen kernel and the target,

$$\begin{aligned}A\left(\boldsymbol{K}_\theta, \boldsymbol{K}^*\right) &= \frac{\langle \boldsymbol{K}_\theta, \boldsymbol{K}^*\rangle_F}{\|\boldsymbol{K}_\theta\|_F \|\boldsymbol{K}^*\|_F} \\ &= \frac{\langle \boldsymbol{K}_\theta, \boldsymbol{K}^*\rangle_F}{M\|\boldsymbol{K}_\theta\|_F},\end{aligned} \tag{B14}$$

where $\boldsymbol{K}_\theta$ is the kernel matrix of κ_θ.

The KTA enjoys theoretical properties such as concentration around its expected value and generalisation [24], and therefore it is indicative of the ability of a kernel to separate classes of data.

Appendix C. Quantum computing notions

C1 Quantum states

The basic logical unit in quantum computing is the qubit. Mathematically speaking, this is a unit-norm vector in the complex 2-dimensional space $\mathbb{C}^2$ defined as a linear combination of two orthogonal basis states, $|0\rangle$ and $|1\rangle$, i.e.,

$$|\psi\rangle = \psi_0|0\rangle + \psi_1|1\rangle, \quad \psi_0, \psi_1 \in \mathbb{C}, \tag{C1}$$

where the $|\cdot\rangle$ notation is used to indicate unit vectors.

As opposed to classical bits, Eq. (C1) shows that a single qubit can be in any complex superposition of the two basis states. However, reading of a quantum state can only happen through a measurement, which will make the qubit collapse to one of the two basis states, $|0\rangle$ or $|1\rangle$. More specificallly, the qubit is measured as $|0\rangle$ with probability $p_0 = \psi_0^2$ and as $|1\rangle$ with probability $p_1 = \psi_1^2$. Since these are the only two possible outcomes, it must be that $p_0 + p_1 = \psi_0^2 + \psi_1^2 = 1$, which explains the unitary norm of the qubit.

Similarly, a state of n qubits is defined as a superposition of basis states that correspond to bitstrings, that is

$$\begin{aligned}&|\psi\rangle = \psi_0|0\ldots0\rangle + \psi_1|0\ldots1\rangle + \cdots + \psi_{N-1}|1\ldots1\rangle, \\ &\psi_0, \psi_1, \ldots, \psi_{N-1} \in \mathbb{C},\end{aligned} \tag{C2}$$

where $N = 2^n$.

The exponential relation between the number of qubits and the number of possible bitstrings speaks for the potential advantage of quantum superposition, which allows multiple classical information states to be processed simultaneously through a quantum algorithm. Quantum superposition is at the heart of fundamental algorithms with proven complexity improvement, such as quantum integer factoring [37] and quantum database search [38].

Nevertheless, the quantum state is inaccessible as readable information and measurement will collapse the wavefunction to only one of the 2^n basis states. Similarly to the single-qubit case, the basis state $|i\rangle$ has probability $p_i = \psi_i^2$ of being measured and

$$\sum_{i=0}^{N-1} p_i = \sum_{i=0}^{N-1} \psi_i^2 = 1. \tag{C3}$$

Quantum states can be prepared by applying unitary transformations to a reference state, such as the all-zero state,

$$|\psi\rangle = U|0\rangle^{\otimes n}, \tag{C4}$$

where U is the generic unitary transformation.

Figure C1 shows a unitary operation as a quantum circuit, i.e., a sequence of single- and two-qubit operations. Here, Hadamard gates H are first applied to every qubit, where

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \\ H|1\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right). \end{aligned} \tag{C5}$$

This first layer of Hadamard gates creates the uniform superposition state

$$\frac{1}{\sqrt{2^n}}\left(|0\ldots0\rangle, |0\ldots1\rangle, \ldots, |1\ldots1\rangle\right), \tag{C6}$$

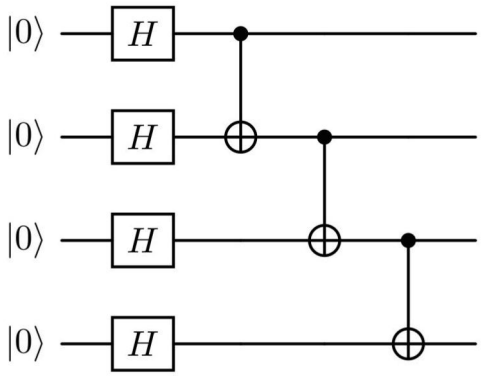

Figure C1 As an example of a quantum circuit, the GHZ state can be prepared by first creating a uniform superposition of all qubits through the Hadamard gate H and then introducing entanglement with CNOT gates in cascade.

where each basis state can be sampled with the equal probability $1/2^n$. This is often a starting point state in many quantum algorithms.

Following, CNOT gates act between neighbouring pairs of qubits as

$$\begin{aligned} &\text{CNOT}|00\rangle = |00\rangle, \quad \text{CNOT}|01\rangle = |01\rangle, \\ &\text{CNOT}|10\rangle = |11\rangle, \quad \text{CNOT}|11\rangle = |10\rangle. \end{aligned} \tag{C7}$$

CNOT gates are used to set qubits in an entangled state, a condition in which any operation on any of the qubits also affects the rest of the state. In particular, the series of CNOT gates creates one of the maximally entangled Greenberger-Horne-Zeilinger (GHZ) states [39], specifically

$$|\text{GHZ}\rangle = \frac{1}{\sqrt{2}}\left(|0000\rangle + |1111\rangle\right). \tag{C8}$$

C2 Quantum embedding

State preparation can be used to embed classical data into quantum states, by mapping the features to a unitary transformation,

$$|\phi(\boldsymbol{x})\rangle = U(\boldsymbol{x})\,|0\rangle. \tag{C9}$$

A complete review of the different types of quantum embeddings is beyond the current scope, and the interested reader is pointed to Ref. [40] for a critical overview.

C3 Quantum kernels

Quantum embeddings are effectively feature maps in the Hilbert space $\mathcal{H} \subseteq \mathbb{C}^{2^n}$. The kernel associated with it computes the overlap between quantum feature vectors in $\mathcal{H}$, that is

$$\kappa(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}') = \left|\langle\phi(\boldsymbol{\varepsilon})\,|\phi(\boldsymbol{\varepsilon}')\rangle\right|^2, \tag{C10}$$

where the braket notation $\langle\cdot|\cdot\rangle$ indicates the inner product between two vectors in $\mathcal{H}$.

By introducing Eq. (C9) in Eq. (C10), the quantum kernel can be rewritten as

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \langle 0|U^{\dagger}(\boldsymbol{x}')\,U(\boldsymbol{x})\,|0\rangle, \tag{C11}$$

which shows that the quantum kernel can be computed as the probability of the all-zeros state, after applying the direct embedding for $\boldsymbol{x}$ and the reversed embedding for $\boldsymbol{x}'$.

Appendix D. Classical and quantum SVM comparison on different classification metrics

Table D1 Comparison of five different classification scores between the RBF-SVM and the trained HE2W6D3 quantum kernel SVM. The classifiers were trained with increasing fractions of the training dataset. Notice that the RBF-SVM problem used $C = 10^7$, while the HE2W6D3 quantum kernel SVM used $C = 10^4$, which is the highest C value before the occurrence of convergence issues

N_{train}	Accuracy	Jaccard index	Precision	Recall	Specificity
RBF kernel					
156	0.694	0.627	0.784	0.759	0.795
313	0.750	0.708	0.835	0.824	0.838
470	0.788	0.751	0.886	0.832	0.895
627	0.788	0.750	0.893	0.825	0.903
784	0.838	0.813	0.939	0.859	0.944
940	0.824	0.797	0.915	0.861	0.921
1097	0.848	0.827	0.938	0.874	0.943
1254	0.878	0.866	0.943	0.913	0.945
1411	0.873	0.860	0.937	0.912	0.939
1568	0.882	0.869	0.955	0.906	0.958
HE2W6D3 kernel					
156	0.699	0.626	0.804	0.739	0.823
313	0.731	0.675	0.835	0.780	0.846
470	0.756	0.708	0.858	0.802	0.869
627	0.779	0.741	0.878	0.826	0.887
784	0.795	0.762	0.892	0.839	0.900
940	0.794	0.757	0.897	0.829	0.907
1097	0.805	0.773	0.902	0.844	0.910
1254	0.797	0.765	0.884	0.849	0.891
1411	0.814	0.785	0.911	0.851	0.917
1568	0.818	0.790	0.912	0.855	0.919

Conflict of interest *On behalf of all authors, the corresponding author states that there is no conflict of interest.*

Author contributions *Giorgio Tosti Balducci developed the methodology, curated the data, performed the formal analysis and investigation, developed the software, processed the data, created the visualization, and wrote the first draft of the manuscript. Boyang Chen developed the methodology, performed the supervision, acquired the funding, and reviewed & edited the manuscript. Matthias Möller and Marc Gerritsma performed the supervision and reviewed & edited the manuscript. Roeland De Breuker performed the supervision, acquired the funding, and reviewed & edited the manuscript. Boyang Chen and Giorgio Tosti Balducci revised and edited the final version.*

References

1 J. M. Whitney, and R. J. Nuismer, Stress fracture criteria for laminated composites containing stress concentrations, J. Compos. Mater. **8**, 253 (1974).
2 P. P. Camanho, G. H. Erçin, G. Catalanotti, S. Mahdi, and P. Linde, A finite fracture mechanics model for the prediction of the open-hole strength of composite laminates, Compos. Part A-Appl. Sci. Manuf. **43**, 1219 (2012).
3 G. Catalanotti, R. M. Salgado, and P. P. Camanho, On the stress intensity factor of cracks emanating from circular and elliptical holes in orthotropic plates, Eng. Fract. Mech. **252**, 107805 (2021).
4 C. Furtado, A. Arteiro, M. A. Bessa, B. L. Wardle, and P. P. Camanho, Prediction of size effects in open-hole laminates using only the Young's modulus, the strength, and the $\Re$-curve of the 0° ply, Compos. Part A-Appl. Sci. Manuf. **101**, 306 (2017).
5 P. P. Camanho, P. Maimí, and C. G. Dávila, Prediction of size effects in notched laminates using continuum damage mechanics, Compos. Sci. Tech. **67**, 2715 (2007).
6 S. R. Hallett, B. G. Green, W. G. Jiang, and M. R. Wisnom, An experimental and numerical investigation into the damage mechanisms in notched composites, Compos. Part A-Appl. Sci. Manuf. **40**, 613 (2009).
7 F. P. van der Meer, L. J. Sluys, S. R. Hallett, and M. R. Wisnom, Computational modeling of complex failure mechanisms in laminates, J. Compos. Mater. **46**, 603 (2011).
8 B. Y. Chen, T. E. Tay, P. M. Baiz, and S. T. Pinho, Numerical analysis of size effects on open-hole tensile composite laminates, Compos. Part A-Appl. Sci. Manuf. **47**, 52 (2013).
9 F. P. van der Meer, C. Oliver, and L. J. Sluys, Computational analysis of progressive failure in a notched laminate including shear nonlinearity and fiber failure, Compos. Sci. Tech. **70**, 692 (2010).
10 B. Y. Chen, T. E. Tay, S. T. Pinho, and V. B. C. Tan, Modelling the tensile failure of composites with the floating node method, Comput. Methods Appl. Mech. Eng. **308**, 414 (2016).
11 C. Soutis, N. A. Fleck, and P. A. Smith, Failure prediction technique for compression loaded carbon fibre-epoxy laminate with open-holes, J. Compos. Mater. **25**, 1476 (1991).
12 Z. C. Su, T. E. Tay, M. Ridha, and B. Y. Chen, Progressive damage modeling of open-hole composite laminates under compression, Compos. Struct. **122**, 507 (2015).
13 R. Higuchi, S. Warabi, A. Yoshimura, T. Nagashima, T. Yokozeki, and T. Okabe, Experimental and numerical study on progressive damage and failure in composite laminates during open-hole compression tests, Compos. Part A-Appl. Sci. Manuf. **145**, 106300 (2021).
14 C. Bisagni, and L. Lanzi, Post-buckling optimisation of composite stiffened panels using neural networks, Compos. Struct. **58**, 237 (2002).
15 M. A. Bessa, and S. Pellegrino, Design of ultra-thin shell structures in the stochastic post-buckling range using Bayesian machine learning and optimization, Int. J. Solids Struct. **139-140**, 174 (2018).
16 Z. Zhang, Z. Zhang, F. Di Caprio, and G. X. Gu, Machine learning for accelerating the design process of double-double composite structures, Compos. Struct. **285**, 115233 (2022).
17 X. Liu, S. Tian, F. Tao, and W. Yu, A review of artificial neural networks in the constitutive modeling of composite materials, Compos. Part B-Eng. **224**, 109152 (2021).
18 N. Zobeiry, J. Reiner, and R. Vaziri, Theory-guided machine learning for damage characterization of composites, Compos. Struct. **246**, 112407 (2020).
19 J. Reiner, R. Vaziri, and N. Zobeiry, Machine learning assisted characterisation and simulation of compressive damage in composite laminates, Compos. Struct. **273**, 114290 (2021).
20 C. Furtado, L. F. Pereira, R. P. Tavares, M. Salgado, F. Otero, G. Catalanotti, A. Arteiro, M. A. Bessa, and P. P. Camanho, A methodology

to generate design allowables of composite laminates using machine learning, Int. J. Solids Struct. **233**, 111095 (2021).
21 C. Cortes, and V. Vapnik, Support-vector networks, Machine Learn. **20**, 273 (1995).
22 B. G. Green, M. R. Wisnom, and S. R. Hallett, An experimental investigation into the tensile strength scaling of notched composites, Compos. Part A-Appl. Sci. Manuf. **38**, 867 (2007).
23 Dassault Systèmes, ABAQUS Finite Element Analysis Software (Dassault Systèmes, Providence, 2022).
24 T. Wang, D. Zhao, and S. Tian, An overview of kernel alignment and its applications, Artif. Intell. Rev. **43**, 179 (2015).
25 B. Schoelkopf, and A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (MIT Press, London, 2019).
26 T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P. J. H. S. Derks, P. K. Faehrmann, and J. J. Meyer, Training quantum embedding kernels on near-term quantum computers, Phys. Rev. A **106**, 042431 (2022).
27 O. Kyriienko, and E. B. Magnusson, Unsupervised quantum machine learning for fraud detection, (2022), doi: 10.48550/arXiv.2208.01203.
28 M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103**, 032430 (2021).
29 D. P. Kingma, and J. Ba, Adam: A method for stochastic optimization, (2014), doi: 10.48550/arXiv.1412.6980.
30 J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).
31 K. Scheinberg, An efficient implementation of an active set method for SVMs, J. Mach. Learn. Res. **7**, 2237 (2006).
32 C. Sentelle, G. C. Anagnostopoulos, and M. Georgiopoulos, in A fast revised simplex method for SVM training: Proceedings of the 2008 19th International Conference on Pattern Recognition, Tampa, 2008, pp. 1-4.
33 S. Setoodeh, M. M. Abdalla, and Z. Gürdal, Design of variable-stiffness laminates using lamination parameters, Compos. Part B-Eng. **37**, 301 (2006).
34 A. Miroszewski, J. Mielczarek, G. Czelusta, F. Szczepanek, B. Grabowski, B. Le Saux, and J. Nalepa, Detecting clouds in multispectral satellite images using quantum-kernel support vector machines, IEEE J. Sel. Top. Appl. Earth Observations Remote Sens. **16**, 7601 (2023).
35 G. Gentinetta, D. Sutter, C. Zoufal, B. Fuller, and S. Woerner, in Quantum kernel alignment with stochastic gradient descent: Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), Bellevue, 2023, pp. 256-262.
36 Dassault Systèmes, ABAQUS Analysis User Manual. Version 2022 edn (Dassault Systèmes, Providence, 2022).
37 P. Shor, in Algorithms for quantum computation: Discrete logarithms and factoring: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, 1994.
38 L. K. Grover, in A fast quantum mechanical algorithm for database search: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, New York, 1996, pp. 212-219.
39 D. M. Greenberger, M. A. Horne, and A. Zeilinger, Going beyond Bell's theorem, in Bell's Theorem, Quantum Theory and Conceptions of the Universe (Springer Netherlands, Dordrecht, 1989), pp. 69-72.
40 M. Schuld, and F. Petruccione, Machine Learning with Quantum Computers (Springer International Publishing, Cham, 2021).

用基于经典和量子核的支持向量机预测开孔层合板的失效

Giorgio Tosti Balducci, Boyang Chen, Matthias Möller, Marc Gerritsma, Roeland De Breuker

摘要 复合材料开孔失效建模是一项复杂的任务, 需同时考虑高度非线性的材料属性与相互影响的多重失效模式. 该材料的数值建模一般是基于有限元法, 需要在高保真度和计算成本间权衡. 为了弥补这一缺陷, 近期的研究开始探索利用机器学习来预测开孔复合材料试件的强度. 本文提出了一个基于数据的机器学习模型, 从分类的角度来处理开孔复合材料的失效问题, 并展示了如何训练模型来学习平面内载荷下开孔复合材料板的最终失效包线. 为此, 我们利用支持向量机(SVM)解决分类问题, 并通过更改SVM核函数来测试不同的分类器. 基于核的SVM的灵活性还使我们能够将最近开发的量子核集成到我们的算法中, 并将其与传统的径向基函数(RBF)核进行比较. 最后, 基于核-目标对齐优化确定自由参数，以区分安全状态与失效状态. 结果表明, RBF的分类准确率高于90%, 量子核次之.