TU Delft

Delft University of Technology

Document Version
Final published version

Licence
CC BY

Citation (APA)
Hegde, A. V., Zarouchas, D., & Caglar, B. (2026). Accelerating curing simulations of composites using Graph Neural Networks. *Composites Part A: Applied Science and Manufacturing*, *211*, Article 110186. https://doi.org/10.1016/j.compositesa.2026.110186

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

*This work is downloaded from Delft University of Technology.*

Composites: Part A 211 (2026) 110186

Contents lists available at ScienceDirect

Composites Part A

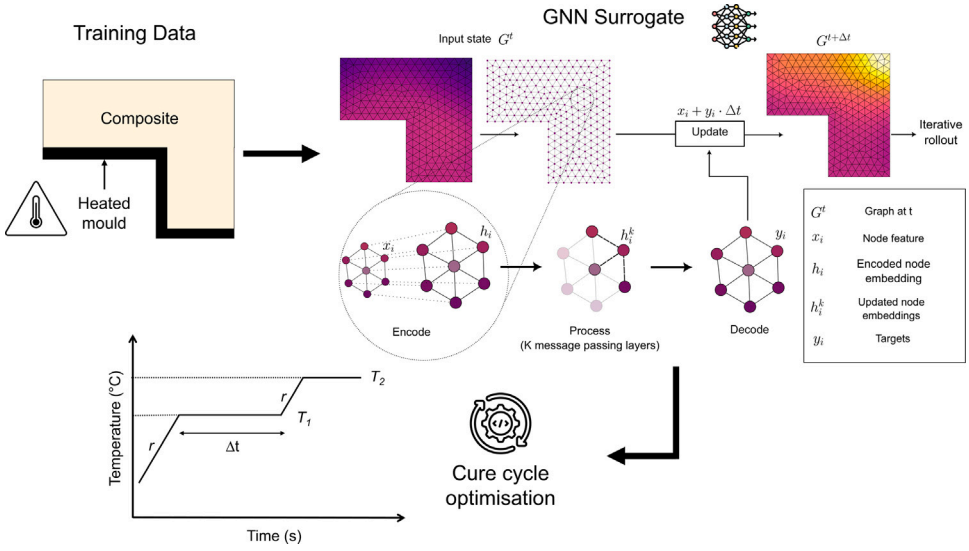
journal homepage: www.elsevier.com/locate/compositesa

# Accelerating curing simulations of composites using Graph Neural Networks

Ashish V. Hegde, Dimitrios Zarouchas, Baris Caglar *

*Aersopace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, South-Holland, The Netherlands*

## GRAPHICAL ABSTRACT




## ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.20177475, https://github.com/camp-lab-tud/GNN-cure-simulation

*Keywords:*
Composite curing
Cure cycle optimisation
Graph Neural Networks
Thermo-chemical modelling
Process optimisation

## ABSTRACT

In thermoset composite manufacturing, process simulation is essential for identifying optimal cure cycles to avoid temperature overshoots, non-uniform curing, or overly conservative cure cycles. Deep Learning based techniques have emerged as promising tools to accelerate modelling compared to conventional numerical methods. This work presents a Graph Neural Network with a MeshGraphNet style architecture applied to the 2D thermochemical modelling of the curing of carbon-fibre reinforced composites. The results of the model in predicting the transient evolution of temperature and degree of cure for rectangular and L-shaped cross-sections under two-dwell cure cycles are presented, showing that the predictions agree closely with finite element results. The areas where the model's predictions struggle are identified. When coupled with a genetic algorithm for cure cycle optimisation to reduce peak exothermic temperature and cure time, the surrogate reduces runtime by about 96.3% compared to finite element simulations. Additionally, the model's flexibility to anisotropic conductivity, as well as other geometries such as T- and U-shapes when such variability is included in the training, is presented. The limitations regarding transferability to other boundary conditions are also identified while recognising directions for future development towards more generalisable surrogates for manufacturing processes.

* Corresponding author.
*E-mail address:* B.Caglar@tudelft.nl (B. Caglar).

https://doi.org/10.1016/j.compositesa.2026.110186
Received 19 May 2026; Received in revised form 8 August 2026; Accepted 10 August 2026
Available online 21 August 2026
1359-835X/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Curing is a crucial phase in the processing of fiber-reinforced composites with a thermosetting matrix, during which the polymer molecules in the resin crosslink, thereby hardening the material and setting into the final shape with the desired properties. Several processing routes can be employed for curing by exposing the uncured part to elevated temperatures and pressures, such as the autoclave or liquid composite moulding techniques [1]. Typically, a controlled temperature profile, also referred to as the cure cycle, is applied to the part to promote polymerisation, and thus obtaining the desired mechanical performance from the part depends on a successful curing stage.

The exothermic nature of the polymerisation reactions, coupled with the composite's lower thermal conductivity, causes temperature heterogeneity within the parts. Non-optimal cure cycles will cause process-induced defects like resin degradation, porosity, warpage of the part due to differential thermal expansion, and generate residual stresses [2–4]. This is especially challenging when manufacturing thicker parts, typically greater than 10 mm thickness, which trap internal heat and can have large thermal gradients [5]. Not having a high enough curing temperature will result in under-curing or extended development times due to inefficiently long cure cycles. Besides minimising process-induced defects, the selection of cure cycles directly impacts production throughput and costs as the curing stage accounts for a significant portion of total manufacturing time and energy consumption, especially for large parts such as wings, wind turbine blade sections and spars. To comply with strict performance requirements and tight development schedules, it would be inefficient and expensive to perform trial-and-error testing for identifying the optimal cure cycle. Therefore, it is essential to accurately simulate the curing process and optimise the cure cycle during the process design phase.

The thermochemical nature of the curing process is well established and governed by partial differential equations (PDEs) that couple reaction kinetics and heat transfer. Solving this system of PDEs allows one to estimate the temperature and cure evolution within the part. Since a closed-form solution is intractable, numerical approximation methods such as the Finite Element Method (FEM) are employed. To describe them crudely, they involve discretising the domain in space and time, approximating a piecewise polynomial solution on each element, and assembling them to obtain the global solution [6]. Currently, the curing stage can be modelled using FEM in a relatively user-friendly manner with special modules and subroutines in commercially available software (e.g., COMSOL, Abaqus, Ansys) [7]. However, these numerical methods incur exorbitant computational costs as the scale and complexity of the model increase. Optimisation of cure cycles involves solving the thermochemical model for a large number of candidate process parameters. Therefore, it becomes impractical to routinely utilise high-fidelity FEM simulations with optimisation techniques such as genetic algorithms. This is particularly restrictive in exploring large design spaces with multiple dwell temperatures in the cure cycle.

Machine Learning (ML) methods, particularly in Deep Learning, have emerged as promising surrogate models to accelerate simulations. Neural Networks (NN) can learn an approximating function that maps input data to the output, including both linearities and non-linearities, with the advantage of being fully differentiable and accurate [8]. Training a network involves minimising a loss function that represents the error between the predictions and the actual target value. So, a trained NN can directly approximate a PDE and estimate the solution for new inputs at a fixed computational cost. There has already been some research on using deep learning in scenarios relevant to composites manufacturing, such as estimating permeability [9] and predicting flow fields in fibrous microstructures [10]. Ref. [11] provides a review of the research on using ML in different processing stages, namely, forming of fibres, flow through fibrous structures, automated fibre placement, and autoclave processing. In composite curing, NNs have been shown to learn from FEM simulations to predict the temperature profile and cure degree [12–14]. A limitation of image-based methods using convolutional neural networks, as in [14], is that they require the input to be a regular grid. This makes them unsuitable for complex geometries and less robust. With a focus on real-time process control, Humfeld et al. proposed a framework coupling a neural network for thermochemical modelling with multi-objective optimisation of the cure temperature in an autoclave, using real-time thermocouple data as input to predict possible boundary conditions and an NN to identify an optimal cure cycle [15]. This was demonstrated on a simplified 1D thermo-chemical model, which limits its generalisability to full-scale 3D industrial parts. While all these works demonstrate near FEM accuracy from the learned models and demonstrate their effectiveness as a surrogate that accelerates simulations, they use different structured architectures trained on 1D data, which limits their ability to scale to spatially complex 2D or 3D domains and restricts flexibility to variable geometries and boundary conditions.

The models used in the different approaches discussed above learn a mapping between the input and output data without actually learning the physics that drives the system. Recently, physics-informed machine learning has been explored as a means to guide the models to obey the physics of the underlying system by incorporating a residual derived from the governing differential equation in the loss function when training the model [16]. Niaki et al. demonstrated this methodology in modelling the curing behaviour of a composite by using residuals derived from the strict form of the PDE governing curing [17]. While they inherently follow the physics more closely, these models are challenging to develop because boundary conditions are often tricky to implement and require explicit handling. They are also computationally expensive to train, as the loss function can become very complex for multi-physics phenomena.

Since traditional neural networks are limited to operating on regular structured data, Graph Neural Networks (GNNs) have emerged as a state-of-the-art family of NNs to address this limitation [18]. They can operate on irregularly structured graph representations and make informed predictions by accounting for the relationships between objects in the data, making them a promising candidate for learning physics simulations, which involve spatial dependencies in the domain. In fact, their effectiveness in modelling dynamic systems by making predictions in an autoregressive manner has been demonstrated by the MeshGraphNet architecture proposed by Pfaff et al. [19]. They demonstrated how a meshed domain, obtained by finite element discretisation, can be represented directly as a graph. A GNN trained on these graph representations can learn to predict the system's dynamic evolution. MeshGraphNet was demonstrated on multiple complex physics problems, including fluid flow around an object and the movement of a cloth. The main advantage of such an approach is the flexibility to use different mesh shapes and resolutions, as well as meshes with different sizes in the same domain. There is also research on making this approach physics-informed by applying it to 2D heat transfer problems [20].

Despite recent advances in graph-based surrogate modelling for physical systems, their application in modelling the coupled thermo-chemical curing process of fibre-reinforced composites remains unexplored. In particular, the ability of such models to accurately learn the strongly non-linear exothermic behaviour associated with polymerisation reactions in composite structures has not been investigated.

In this work, we apply a MeshGraphNet-inspired architecture to model the curing of a carbon-fibre-reinforced composite material using the commonly used aerospace-grade thermosetting resin RTM6. We demonstrate the model's performance on this thermo-chemical system with coupled partial differential equations governing the physics. We study the model's ability to learn the non-linear temperature history through the thickness of the material in both rectangular and L-shaped cross sections, subjected to a prescribed cure cycle on one boundary and natural air convection on the other boundaries. We then highlight

the speed of using a trained model for simulations by utilising it as a surrogate in a multi-objective optimisation of the cure cycle using a genetic algorithm, aiming to minimise both temperature overshoot and curing time. We also explore the generalisation capability of the model to unseen T- and U-shaped domains composed of the primitive shapes seen by the model during training. After reporting the strengths and weaknesses of the model, we further populate the dataset by introducing anisotropic conductivity and introducing simulation data from T- and U- shaped geometries. We then report the improved performance of the model after training on the broadened dataset, and also the limitations in terms of handling unseen boundary conditions. These limitations can be addressed with a broader version of the model that can incorporate not only geometrical variability but also material variability and different boundary conditions.

## 2. Thermochemical modelling of the curing process

The heat transfer equation governing the curing process is given by:

$$\rho_c C_{pc} \frac{\partial T}{\partial t} = \nabla \cdot \left(k_c \nabla T\right) + \dot{Q} \tag{1}$$

where $\rho_c$, $C_{pc}$, and $k_c$ are the density, specific heat capacity and thermal conductivity of the composite material. $T$ is the temperature of the material, $t$ is time and $\dot{Q}$ is a source term describing the internal heat generation in the material due to the exothermic polymerisation reactions. The subscript $c$ refers to the material's homogenised properties as dictated by the rule of mixtures.

The internal heat generation can be considered proportional to the reaction rate and is expressed as:

$$\dot{Q} = v_r \rho_r H_r \frac{d\alpha}{dt} \tag{2}$$

where the subscript $r$ refers to the resin, $v_r$ is the volume fraction, $H_r$ is the heat of reaction generated by the resin during cure, and $\frac{d\alpha}{dt}$ is the reaction rate. The degree of cure $\alpha \in (0,1)$ represents the fraction of material that has cured. The reaction rate is typically described as a function of the instantaneous temperature and degree of cure, and can be described by different cure kinetics [21].

### 2.1. Cure kinetics

For our study, we consider a carbon fibre-reinforced composite consisting of G1157 carbon fibres and RTM6 resin, both from Hexcel Composites. The material choice and properties are based on [22]. The cure kinetics for this resin can be described by a modified Kamal–Sourour equation, which accounts for a transition from chemically controlled to diffusion-controlled kinetics since fewer reaction partners are available for cross-linking as the cure progresses [23]:

$$\frac{d\alpha}{dt} = f_D \cdot \left(k_1 + k_2 \alpha^m\right) \cdot (1-\alpha)^n \tag{3}$$

where $k_1$ and $k_2$ are the reaction rate constants, $m$ and $n$ are the reaction orders, and $f_D$ is the reaction rate limiter based on the final degree of cure $\alpha_f$ given by:

$$f_D = \left[\frac{2}{1-\exp\left(\frac{\alpha-\alpha_f}{b}\right)} - 1\right] \tag{4}$$

We use the material properties and cure kinetics parameters listed in Table 1, adapted from the characterisations in Refs. [22,23].

### 2.2. Cure simulation

Training data for the GNN surrogate were generated via finite-element cure simulations in COMSOL. The study is restricted to a computational approach, and the finite element results serve as the basis for comparison with the surrogate's predictions. Fig. 1 provides a schematic of the simulation setup and the two-dwell cure cycle used for the simulations.

**Table 1**
Material properties and cure kinetics parameters.

| Parameter | Value |
|---|---|
| $H_r$ | 419 J g$^{-1}$ |
| $A_1$ | 75000 s$^{-1}$ |
| $A_2$ | 21667 s$^{-1}$ |
| $E_1$ | 74690 J mol$^{-1}$ |
| $E_2$ | 58370 J mol$^{-1}$ |
| $m$ | 1.2 |
| $n$ | 1.15 |
| $b$ | 0.05 |
| $k_c$ | 0.4 W m$^{-1}$ K$^{-1}$ |
| $\rho_f$ | 1760 kg m$^{-3}$ |
| $\rho_r$ | 1110 kg m$^{-3}$ |
| $\rho_c$ | 1500 kg m$^{-3}$ |
| $C_{pc}$ | 1360 J kg$^{-1}$ K$^{-1}$ |

Herein, we restrict ourselves to 2D simulations on flat-panel and L-shaped geometries and demonstrate the capabilities and limitations of GNNs in terms of generalisability to unseen geometries in Appendix D. Since the primary focus of this study is to investigate the ability of the GNN surrogate to model the non-linear temperature evolution and degree of cure across the part thickness, simplifications are made to keep the training data relatively simple and the data generation computationally tractable. In the initial stage of the model development, uniform conductivity is considered in all directions. After studying the model capabilities and limitations with uniform conductivity, the performance of the model is assessed when more complexity is introduced into the simulation by assuming a quasi-isotropic layup with different longitudinal and transverse conductivities.

For curing, boundary conditions encountered in Vacuum Assisted Resin Transfer Moulding (VARTM) are considered. VARTM is a popular manufacturing process for large, complex components, making it ideal for the aerospace and wind turbine industries [24]. The composite's thermal properties are assumed to be independent of the degree of cure. Properties are considered homogenised using the rule of mixtures. Resin flow effects are neglected, assuming that curing begins after the mould filling stage is complete. The tool is not modelled, and the thermal resistance at the tool-part interface is neglected. The study is also restricted to temperature and cure degree prediction, as it does not consider cure-induced shrinkage in the FEM model, and uses peak temperature as a proxy for cure-induced shape changes following the approach in Ref. [25].

Since curing occurs on a hot tool, a prescribed time-dependent boundary condition corresponding to the cure cycle is applied to the tool side. The fibre and resin are assumed to be at a temperature of 120 °C at the beginning of cure. Natural air-convection boundary condition is applied to the vacuum bag side as shown in Fig. 1. A constant natural convection coefficient of 5 W m$^{-2}$ K$^{-1}$ is assumed on the vacuum bag side, representative of typical passive cooling, natural convection boundary conditions. The cure cycle is parametrised by first dwell temperature $T_1$, dwell duration $\Delta t$, second dwell temperature $T_2$, and ramp rate $r$, as shown in 1(b). The standard cure profile comprises a ramp at 1 °C min$^{-1}$ from 120 °C to 160 °C, followed by a dwell at 160 °C for 4500 s, a second ramp at 1 °C min$^{-1}$ to 180 °C, and a dwell at this temperature till curing is complete. The second dwell temperature is typically the post-cure temperature, which must be higher than the glass transition temperature reached during the first dwell. The study in [26] serves as the basis for typical initial and boundary conditions.

The governing equations were solved using a transient finite element study in COMSOL Multiphysics. The spatial domain is discretised with triangular elements, and an implicit backward differentiation formula (BDF) scheme with adaptive time stepping and relative tolerance of 0.001 is used for time integration. An implicit formulation is adopted due to the stiffness of the governing equations from strongly non-linear

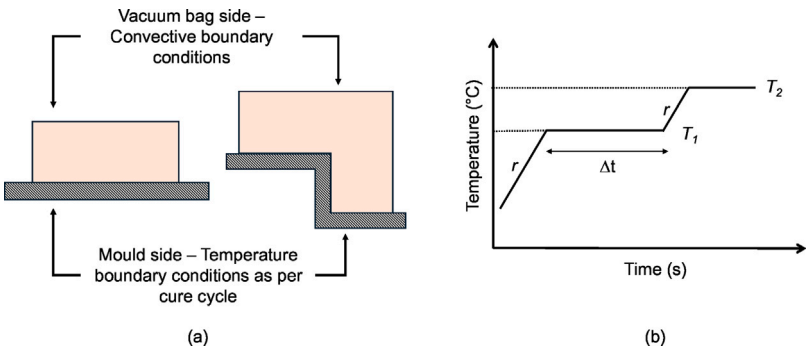



**Fig. 1.** Cure simulation setup (a) Geometries and boundary conditions (b) General parametrised cure cycle.

cure kinetics and heat generation. A mesh independence study was conducted to ensure that the solution is not affected by numerical discretisation. When storing the results, the nodal values of temperature and cure degree are recorded every 100th second. This means that simulations in the training data have a fixed timestep of 100 s. This ensures that the training data set is not too large, since curing times range into the hundreds of minutes.

## 3. Building the GNN surrogate

Our model is based on MeshGraphNet utilising an encode-process-decode architecture [19,27]. Conventional neural-network architectures such as convolutional neural networks are limited by the requirement of structured inputs while graph-based networks allow the modelling of meshes with non-uniform sizes. The MeshGraphNet architecture for GNNs has been demonstrated on dynamic simulations such as flow of fluids around an obstacle and was thus chosen as the starting point. Herein, the architecture is adapted for the coupled heat-transfer and cure kinetics multiphysics modelling of composite curing. The architecture choice allows for operating on unstructured meshes, accommodating geometric variability, and propagating information representing the local spatial interactions which are fundamental for the heat transfer and cure evolution. The construction of the graph features, targets to be predicted, and the loss function are described in the following sections.

The overall workflow involves compiling training data from FEM cure simulations, representing the meshed domain as graphs, and subsequently using these graphs with a GNN to estimate the nodal evolution of temperature and cure degree over time. The following sections describe these steps in detail.

### 3.1. Dataset generation and pre-processing

The training data consists of FEM cure simulations of L-shaped and rectangular cross sections with varying thicknesses as described in Section 2.2. Different cure cycles are applied by combinations of varying dwell temperatures, ramp rates, and dwell durations. Table 2 provides the range of values for each of the simulation parameters that are varied. The parameter bounds are based on the standard cure cycle for RTM6 as reported in literature [22]. The limits are chosen to cover a range of processing conditions, including slower cure rates as commonly used for reducing exothermic temperatures and thermal gradients in thicker parts, and faster cycles as used for reducing processing times for thinner parts. The dwell duration range is chosen to have a wide range of degrees of cure after the first dwell ends.

To avoid running a large number of simulations for every parameter combination, and to explore our model's ability to generalise within the parameter space without requiring much training data, we adopt a Sobol sampling strategy. The Sobol sampling ensures that the parameter space is uniformly sampled by the desired number of points. 256 simulations each for the rectangular and L-shaped domains within the ranges mentioned in Table 2 are generated, with an additional 64 simulations in the higher thickness regions to expose the model to more nonlinear thermal responses. The choice of these parameters was made to expose the model to both strongly non-linear temperature overshots and conservative cycles with less temperature gradient within the thickness. In total, the dataset spans 576 simulations, covering both conservative and highly exothermic cure regimes. For each of these, the nodal values of temperature and degree of cure are stored every 100 s. The simulations are then split into training, validation, and test sets with a stratified, shuffled 70-15-15 split. This results in a training dataset of 52,163 graphs spanning 403 simulation cases, with the remaining simulations compiled in the validation and test sets.

**Table 2**
Thickness and cure cycle parameter bounds of simulations in training data.

| Simulation parameter | Range |
|---|---|
| Thickness | 3–30 mm |
| First dwell temperature $T_1$ | 145–175 °C |
| First dwell duration $\Delta t$ | 1800–10800 s |
| Second dwell temperature $T_2$ | 180–210 °C |
| Ramp rate $r$ | 1–3 °C min$^{-1}$ |

When presenting the results, Section 4.4 showcases results for a model trained with a dataset that also includes T and U-shaped cross sections. This dataset was created by adding 256 simulations each for T and U shapes to the previously described dataset. It was motivated by the analysis of the model's performance to shapes outside the training distribution (as described in Appendix D), and it served to test the hypothesis that a single model could be applicable to multiple shapes provided the necessary geometric variability was included during training.

#### 3.1.1. Graph structure

The meshed domains and results as generated in Section 3.1 need to be represented as graphs for use with the GNN. A graph is a representation of a set of objects and their interconnection, in the form of nodes and edges, respectively. Formally, a graph with a set of nodes $\mathcal{V}$ and connecting edges $\mathcal{E}$ is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The edge $(u, v) \in \mathcal{E}$ connects two nodes $u \in \mathcal{V}$ and $v \in \mathcal{V}$.

The PyTorch Geometric package [28] was used to construct graph representations of the meshed domains. For every simulation, the state of the system at a particular time $t$ is represented as the graph $\mathcal{G}^t$, such that the nodes of the graph correspond to the nodes of the mesh and the mesh connectivity defines the edges of the graph. Each node has a feature vector $x_u$ that uniquely represents it. This node feature vector is built using the nodal values of temperature and degree of cure at time $t$, followed by a one-hot encoding of node type. The node type refers to nodes on the temperature boundary, the convection boundary, or in the interior. This results in a 5-dimensional feature vector for each node. The edge features include the 2D position vectors between connected

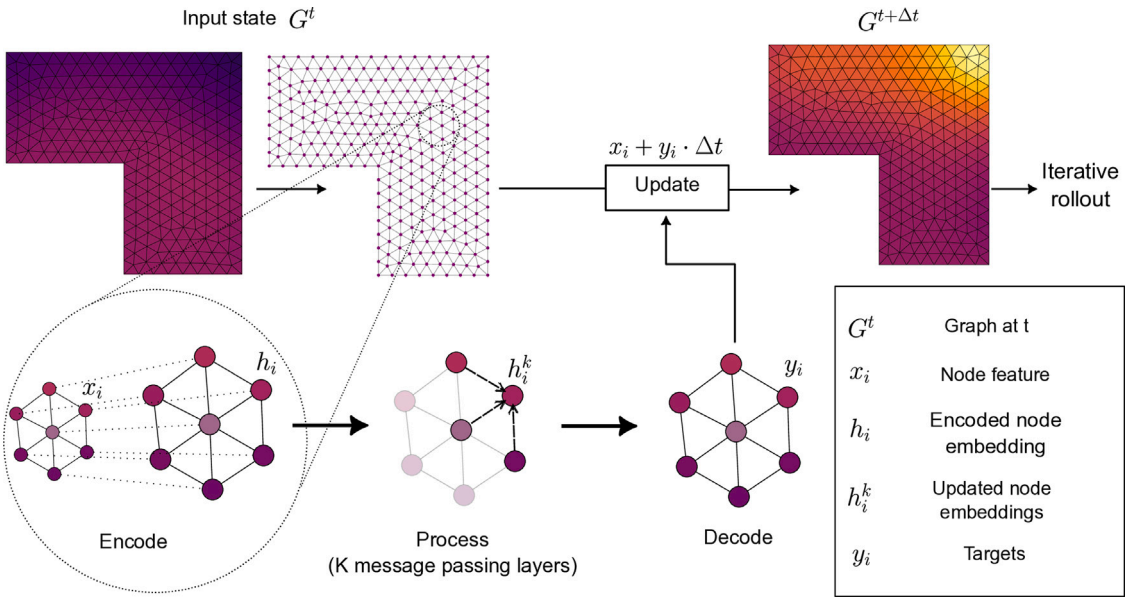



**Fig. 2.** Schematic of the model architecture. The encode-process-decode blocks learn to output the target $y$ at each node given the graph of the input state $G^t$, such that the graph at the next state $G^{t+\Delta t}$ can be reconstructed.

nodes and the distance between them, yielding a 3-dimensional edge feature vector $e_{u,v}$. This edge featurization makes the GNN aware of the mesh structure and flexible to different edge lengths within the domain. Each graph is made aware of the cure cycle, with the aim of coupling later with the genetic algorithm for optimisation, by including the corresponding cure-cycle parameters as global features. This is done by appending the corresponding simulation parameters (as described in Table 2 to the node features, effectively making them 10-dimensional vectors.

Each node also has a target $y$, which is used for training the model. In our case, the targets are the nodal rates of change of temperature $\frac{\Delta T}{\Delta t}$ and degree of cure $\frac{\Delta \alpha}{\Delta t}$. This representation thus facilitates learning the node targets $y$ from the input $G^t$. Predicting two target rates this way also allows the usage of a single model for both temperature and degree of cure prediction.

During inference, each predicted state will serve as input to the next prediction. This will result in an accumulation of errors associated with each predicted output when rolling out the series of predictions for the full simulation. To make the model more robust to this, the node features are perturbed by a small amount of noise [19].

### 3.2. Model architecture

The aim of a MeshGraphNet-style model is to predict the output graph representing the system's state at the next time step, given the graph of the current time step. Each predicted graph can be used in an autoregressive manner to generate a sequence of graphs that represent the system's dynamic evolution. To do this, we rely on encode-process-decode blocks [27]. Fig. 2 provides a schematic overview of the model.

The encoder makes the graph representation more expressive by expanding lower-dimensional feature vectors into higher-dimensional embeddings, ensuring similar nodes are also close to each other in the embedding space. The node encoder is a Multi-Layer Perceptron (MLP) with the Sigmoid Linear Unit (SiLU) activation function [29] to learn the mapping of node features $x_u \; v \in \mathcal{V}$ to 128-dimensional vector embeddings $h_u \in \mathbf{R}^d$ such that:

$$h_u = MLP(x_u) \forall v \in \mathcal{V} \tag{5}$$

Edge features are also similarly encoded by the edge encoder into a 128-dimensional embedding.

The processor consists of $K$ message passing layers. During each message-passing iteration, the hidden embedding $h_u^k$ corresponding to node $u$ is updated based on information aggregated from its neighbours in the neighbourhood $\mathcal{N}(u)$. At each iteration $k$, given node $u$ and its edges $(u, v) \in \mathcal{E}$, an MLP calculates a message by mapping the concatenated source node, target node, and edge embeddings to a message. The messages are then aggregated by summation to generate $m_{\mathcal{N}(u)}^k$

$$m_{\mathcal{N}(u)}^k = \sum_{v \in \mathcal{N}(u)} MLP(h_u^{k-1} \oplus h_v^{k-1} \oplus e_{u,v}^k) \tag{6}$$

where $\oplus$ represents vector concatenation.

An update function then combines the message with the previous embedding $h_u^{(k-1)}$ with a residual connection to generate the updated embedding $h_u^k$. After $K$ message passing layers, the node embeddings will be transformed according to information transmitted from its K-hop neighbourhood.

$$h_u^k = h_u^{(k-1)} + MLP(m_{\mathcal{N}(u)}^k \oplus h_u^{(k-1)}) \tag{7}$$

The number of message-passing layers, $K$, in this model was set to 13 after hyperparameter optimisation (see Appendix B), meaning that spatial information is propagated across a distance of 13 connected nodes. This is an important parameter for training GNNs as it determines how far information propagates. Too few layers mean that information from farther away is lost, while too many layers not only increase the computational cost but also run the risk of oversmoothing information because of node embeddings converging to the same value through repeated updates. In this case, 13 message passing layers is enough to span the thickness of the domains in the dataset, ensuring that a node on one side of the mould is informed of the state of the node on the other side.

The decoder is an MLP that maps updated embeddings after message passing, to two output features that represent the rates of change of temperature and degree of cure. Thus, during training the model learns to map input node features to target rates at each node by minimising the training loss. The total training loss is computed as the sum of the Mean Squared Error (MSE) of the two output rates $\frac{\Delta T}{\Delta t}$ and $\frac{\Delta \alpha}{\Delta t}$. Both the MSEs are weighted equally because the targets, as well as all features, were first standardised before training. The standardisation results in the Temperature and $\alpha$ rates being in the same scale. The MSEs of

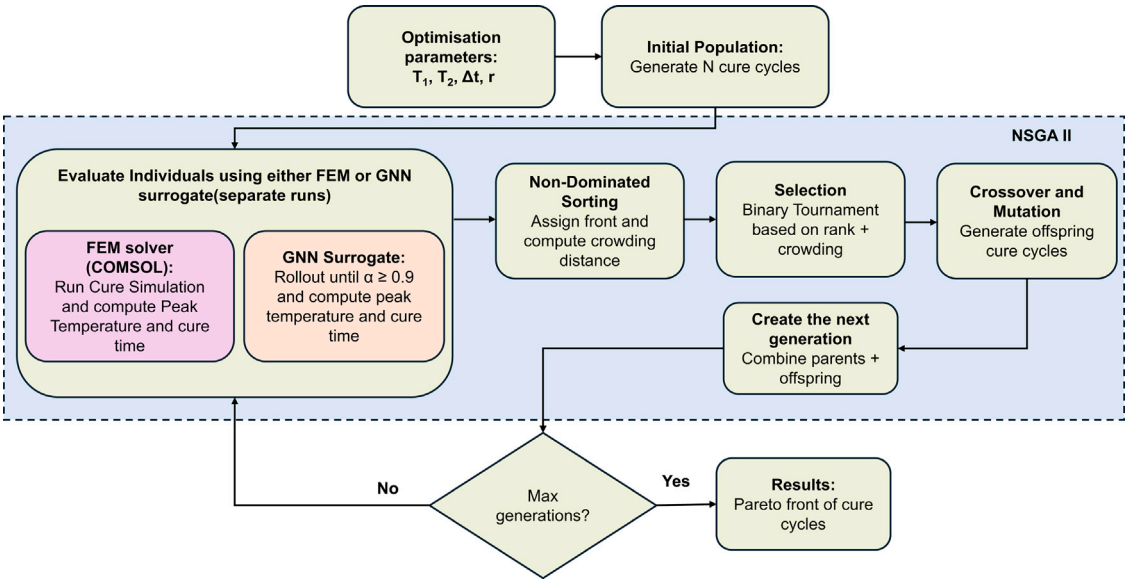



**Fig. 3.** Schematic of the multi-objective optimisation of the cure cycle using the NSGA II algorithm.

**Table 3**
Model hyperparameters.

| Hyperparameter | Value |
|---|---|
| Number of hidden layers in MLPs | 2 |
| Number of message passing layers | 13 |
| Learning rate | 0.0002 |
| Weight decay | 3.6 x $10^{-5}$ |
| Noise scale | 0.04 |

**Table 4**
Design parameter ranges.

| Parameter | Range |
|---|---|
| $T_1$ | 145–180 °C |
| $T_2$ | 175–210 °C |
| $\Delta t$ | 1800–10800 s |
| $r$ | 1–3 °C $min^{-1}$ |

each output hence contribute equally to the total loss despite having different physical magnitudes.

The new graph representing the state at $t + \Delta t$ is obtained by forward integration of the outputs to obtain the new nodal temperatures and degree of cure.

All the encoder, processing, and decoder MLPs utilise the SiLU activation function. The SiLU function was used because it is smooth and non-monotonic, allowing it to capture more complex non-linear patterns in data than the monotonic ReLU [29]. The encoder and processor MLPs also have their outputs normalised by a LayerNorm [30]. The hyperparameters of our model are summarised in Table 3. The chosen architecture results in 1.56 X $10^6$ parameters for the model to learn.

### 3.3. Multi-objective optimisation methodology

Early-stage process design is an area where the process surrogates can be used, as the early-stage results are used for screening and identifying the process bounds before running more detailed and computationally demanding physics-based simulations. To study the capabilities of the trained GNN model in a high-throughput computation scenario, we set up an optimisation task to identify cure cycles that minimise both processing time and the maximum exothermic temperature overshoot within the material. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) provided by the Python package Pymoo [31,32] was used. Two versions of the optimisation code were developed in Python, one that links with the trained GNN model for simulations, and the other that links to COMSOL for FEM simulations. A schematic of the genetic algorithm and solver interface is illustrated in Fig. 3.

The parameters for optimising the cure cycle are the first and second dwell temperatures ($T_1$, $T_2$), the duration of the first dwell ($\Delta t$), and ramp rate (r). The duration of the second dwell is not considered, as the simulations are run until the material reaches a threshold degree of cure of 0.9. For both the GNN and COMSOl models, simulations are stopped when this threshold is reached, and curing is assumed to be complete. The design parameter ranges are summarised in Table 4.

The two objectives to minimise are the total curing time and the peak temperature within the part. Two constraints were imposed to ensure meaningful curing schedules by requiring that the first dwell temperature be at least 5 °C above the initial temperature, and that the second dwell temperature be at least 5 °C above the first. Appendix A describes the simulated binary crossover and mutation used by the NSGA-II algorithm for generating offspring, the termination criteria, and the analysis of convergence.

The same optimisation algorithm is used with both the GNN solver and COMSOL's finite-element solver to identify optimal cure cycles for a 20 mm thick L-shaped geometry. At every iteration, the algorithm generates a set of parameter combinations which serve as candidate solutions for the cure cycle. It requires solving the thermochemical problem described in Section 2, then extracting the objectives: peak temperature and total cure time. Candidate solutions, i.e., the set of parameters that minimise both objectives, are selected to generate the set of parameters for the next iteration. This is repeated until the maximum number of generations is reached, yielding a Pareto front of optimal cure cycle parameters. When using the GNN model to extract objectives, an initial graph is created with boundary conditions of each candidate solution. The GNN predictions are then rolled out until $\alpha > 0.9$ everywhere in the material. When using the FEM model, the GA links to COMSOL using the Python library Mph [33] for running the simulations from a template .mph file by using the simulation parameters in each generation.

## 4. Results and discussion

### 4.1. Model training results

Fig. 4 shows the loss curve after training the model for 500 epochs, which took 11 h and 50 min to complete on an NVIDIA A100 GPU. The

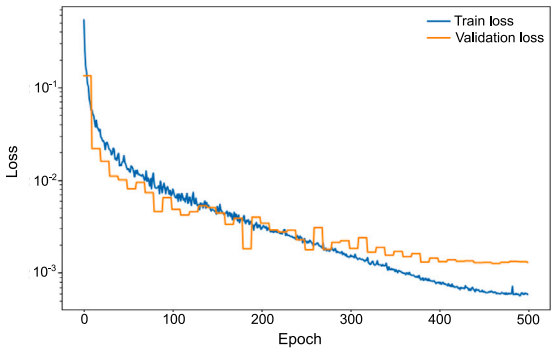



**Fig. 4.** Training and validation losses.

evolution of the MSE loss indicates stable training over 500 epochs. The staircase-like nature of the validation loss is due to it being computed every 10 epochs instead of every epoch, to save computation time. The validation loss starts plateauing at 1 X $10^{-3}$ after ~300 epochs, and the final model saved is the one with the lowest validation loss of 0.001262, at epoch number 460.

The MSE loss indicates the model's single-step prediction error for predicting the combined rates of change of temperature and degree of cure. To evaluate the model's performance in predicting temperature and degree of cure evolution during the curing period, it was tested on all simulations in the test set, which were previously unseen by the model. These results are presented in the parity plot in Fig. 5, which compares the predicted values of temperature and degree of cure against the actual nodal values from FEM simulations at every step for all simulations in the test set. The 45° line $y = x$ indicates perfect predictions. The model achieves a coefficient of determination $R^2$ = 0.9969 and root mean squared error RMSE = 1.4551 °C for temperature predictions. A vast majority of points cluster close to the diagonal, with increased scatter away from the diagonal at temperatures above 250 °C, indicating that the prediction errors are higher at very high temperatures. We note that this temperature range for high errors, as will be discussed in the next section, is beyond the typically encountered temperature range in practice for this resin, and is associated with its degradation [34].

The degree of cure predictions have $R^2$ = 0.9994 and RMSE = 0.0096, indicating good overall performance similar to that of the temperature predictions. More points away from the diagonal lie in the intermediate cure region of ~ 0.4 - 0.7, corresponding to regions where the cure kinetics are highly non-linear and the reaction rates are high. As it will be discussed in the next section, these cases mostly belong to the scenarios where a relatively short first dwell is followed by a second dwell that is at a much higher temperature than the first dwell, in a thick part and the exothermic peak is then captured with a time delay by the model.

### 4.2. Inference performance on unseen cure scenarios

The GNN model rolls out predictions for a simulation in an autoregressive manner. Hence, a low loss value at the end of training, indicating good single-step prediction accuracy, is not necessarily indicative of the model's performance over a lengthy rollout. Fig. 6 shows the model's best predicted case for an L-shaped domain in the test set by comparing the predicted temperatures and cure degrees against FEM results. This simulation case has a thickness of 7 mm, and the cure cycle is defined by a first dwell temperature of 146 °C, dwell duration of 3849 s, second dwell temperature of 199 °C, and ramp rate of 2.5 °C $\text{min}^{-1}$.

Fig. 6(a) and (b) show the spatial distribution of temperature and degree of cure predictions at two different times during the cure, including when the material reaches the highest temperature of 207.3 °C at minute 103.3. The model accurately reconstructs the temperature and degree of cure fields through the thickness of the material, including at the corner of the L shape where the mesh has a slight distortion compared to the rest of the domain. The maximum temperature error throughout the simulation was 1.89 °C, with a mean error of 0.33 °C across all nodes and timesteps. The error in degree-of-cure predictions remains below 0.011, demonstrating the model's ability to track the underlying non-linear cure kinetics. Figs. 6(c) and (d) show the temporal evolution of temperature and the degree of cure for the hottest node. The prediction closely follows the FEM results through all phases of the cure cycle. The point of exothermic temperature overshoot, marked by the dotted red line, is accurately captured in both magnitude and timing. The model also accurately predicts the rapid curing after the 80th minute where the ramp starts and closely follows the FEM results for the degree of cure.

The temporal evolution of temperature and degree of cure at the hottest node for the simulation with the worst prediction is shown in Fig. 7. Very high maximum errors of 164.4 °C for temperature and 0.75 for the degree of cure are observed in this simulation case for an L shape with a thickness of 26 mm. The cure cycle was described by first dwell temperature of 156 °C, dwell duration of 4595 s, second dwell temperature of 206 °C, and ramp rate of 2.5 °C $\text{min}^{-1}$. The location of the hottest node as well as the node with the highest error, highlighted in Fig. 7(a), are close to the convection boundary. The model's predictions closely track the FEM results in the initial phase, as seen in Fig. 7, but the error begins to explode as the temperature and cure rates steeply rise. When taking a closer look at the exotherm region (appendix Figure E.21), it is observed that while the model still captures the overall trend, the individual nodal errors are high because there is a shift in time, resulting in very high errors when comparing the instantaneous predicted values. Snapshot of through-thickness predictions at the instant of maximum error is in the appendix Figure E.20.

The cases where the model struggles to closely follow FEM results are those with a combination of higher thicknesses in the range of 25–30 mm, relatively short first dwell durations and high difference between the first and second dwell temperatures. This is visualised in Fig. 8 (and also E.17 in Appendix E, which shows some extra results for simulations in the test set). This figure was generated by calculating the correlation coefficient between the simulation parameters and the maximum prediction errors for each case in the test set, and visualising their positions in a landscape formed by the parameters most correlated with the prediction errors.

The first dwell period in the cure cycle needs to allow sufficient time for some degree of polymerisation to occur before increasing the mould temperature, such that thermal gradients across the material's thickness and temperature overshoots can be minimised. The temperature overshoot increases with increasing part thickness, due to greater heat release from increased volume and reduced heat dissipation. The combinations of low dwell duration and high difference between dwell temperatures for a thick part thus result in steeper gradients for the model to learn, leading to the errors observed. The approach in this model could be useful for rapid simulations of very thick parts with fast cure cycles, but it is limited in its applicability in its current state. However, it is worth noting that the extreme divergence of the temperature occurs when the peak temperature is beyond 250 °C and such temperatures are not practical for the studied resin whose recommended second dwell temperature is 180 °C [26].

All the worst prediction cases with high errors exhibit the same trend discussed for the result in Fig. 7. The overall trends in the dynamic evolution of the temperature and degree of cure are captured, although with a slight time shift. One reason is that the model is predicting rates of change for the physical quantities, and the number of lower target rates greatly dominates the number of steep rates in the training distribution. The way around this is to incorporate more training data with such steep rates, that is, more simulations of

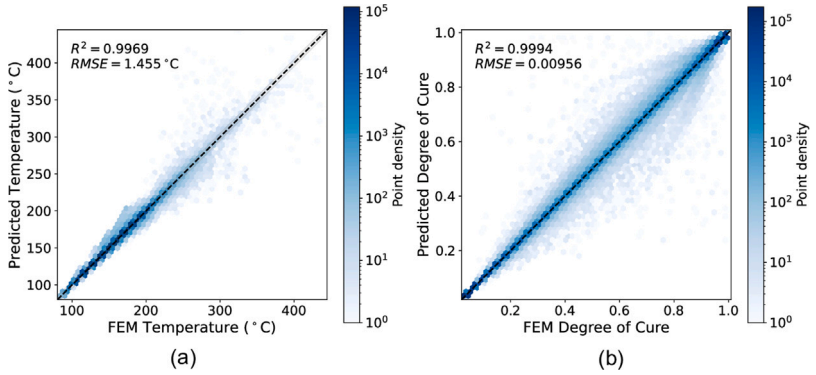



**Fig. 5.** Parity plots of temperature (a) and degree of cure (b) predictions for all simulations in the test set.

very thick components with high dwell temperatures and short dwell periods.

Another likely reason for the deviation in predictions in this regime is the time step size of 100 s taken by the model for prediction. The FEM results during the exotherm can be sampled at smaller time steps to train the model and improve its accuracy. While the instantaneous errors in these extreme cases are high, they are also impractical cases for this resin. For this study, it was deemed unproductive to generate additional training data to improve the performance of the model in extreme cases with very high peak temperatures that are not useful in practice. Also, the FEM results for maximum temperatures during very high overshoots cannot be considered as reliable estimates of the actual temperatures for training, as neither the model nor the FEM simulations do not account for phenomena associated with resin degradation at such extremes, with RTM6 resin degrading at ~360 °C [34]. Hence, the model's current performance was deemed satisfactory for proceeding with testing its potential as an early-stage process design tool for a multi-objective optimisation task.

Regarding the speed of prediction with the trained GNN model, inference times for simulations in the test set were measured on an NVIDIA A100 GPU. The average per-step simulation time is 2.32 ms, and the average time per simulation is 0.32 s. In contrast to the average of 13 s per FEM simulation in COMSOL on a 64-core Intel Xeon CPU, the trained GNN model achieves an average per-simulation speedup of 97.5%. This highlights the potential of using such a model to accelerate simulations for identifying curing temperature bounds in early-stage process design.

To estimate the computational cost, the number of floating point multiplication and addition operations (FLOPs) is a widely used metric. This has to be estimated for COMSOL based on the degrees of freedom and solver type since COMSOL does not directly output FLOP count in its log. Taking an example simulation for an L-shaped domain with 20 mm thickness, for which the number of degrees of freedom is 1678, the estimated number of operations for the COMSOL solver was in the order of $10^9$ FLOPs (see Appendix C for more details).

For the GNN model, the number of FLOPs is dependent on the size of the input graph. For this same simulation case, it was estimated using Pytorch's built in profiling tool that a forward pass required $10^9$ FLOPs. For the whole simulation rollout, the total FLOPS would then be the FLOPs for a forward pass multiplied by the number of rollout steps, leading to $10^{11}$ FLOPs.

Regarding the cost of training the model, the FLOPs per training step are estimated on the order of $10^{10}$ operations. This is calculated for a batch size of 8, considering the number of batches, and the number of operations for forward pass, backward pass, and optimiser update. The model scales linearly with number of nodes and edges, and with the number of message passing layers, while scaling quadratically with the hidden dimension size.

The total count of operations are further increased when considering the dataset generation through COMSOL simulations and points towards two to three orders of magnitude difference between a model that is ready for inference task and a typical finite element simulation. However, it must be noted that FLOPs only count the number of elementary arithmetic operations and do not provide insight into hardware specific details that will affect the runtime. Counting FLOPs also ignores memory access times and communication overheads [35,36]. Modern GPU architectures and tensor cores map efficiently to the dense operations of deep neural networks. Hence, the inferences of the model on the GPU are significantly faster than the FEM simulation on the CPU.

### 4.3. Multi-objective optimisation results

An L-shaped domain with a thickness of 20 mm was considered for multi-objective optimisation of the cure cycle, following the setup described in Section 3.3. A population size of 50 was set for each generation, with the optimisation set to terminate after a maximum of 50 generations. The resulting Pareto fronts of solutions from two runs, one using COMSOL for cure simulations and the other using the trained GNN surrogate, are presented in Fig. 9.

The Pareto front obtained with the GNN surrogate closely matches that generated by FEM simulations. They have an L-shape that reflects the competing nature of the two objectives, with the horizontal region prioritising temperature and the vertical region prioritising curing time. The closeness of the two solutions indicates that the GNN surrogate model can reliably approximate the optimal curing time vs temperature overshoot compromise relevant for process design. The key result is that the optimisation algorithm ran for only 17.3 min using the GNN surrogate for the 2500 simulations, compared to 8 h with COMSOL, thus reducing the optimisation run time by ~96.3%. The GNN surrogate ran on an NVIDIA A100 GPU, while the COMSOL computations ran on an Intel Xeon 4214R CPU. For this study, no explicit parallelisation was used for either running the GNN model on the GPU or for COMSOL on the CPU. Each candidate cure cycle was evaluated sequentially so that the speed-up reflects gains from the surrogate model's inference speed utilising the GPU, and neither approach has an advantage over the other due to parallel execution. When considering the solution closest to the origin in the Pareto front plot as the optimal cure cycle, the cure cycle identified by the GNN surrogate achieves a 30% reduction in the curing time while also reducing the peak temperature within the material by 2 °C.

Overall, the close alignment of the Pareto fronts and the significantly reduced computation time indicate that the GNN surrogate is a computationally efficient alternative of the FEM model in multi-objective cure cycle optimisation. However, it should be noted that the dataset generation and the fact that GNN operations take place in GPUs while FEM simulations are typically on CPUs makes one-to-one comparison more challenging.

In this study, the inspiration comes from the design evolutions of components such as wind turbine blades, where the cross-sectional

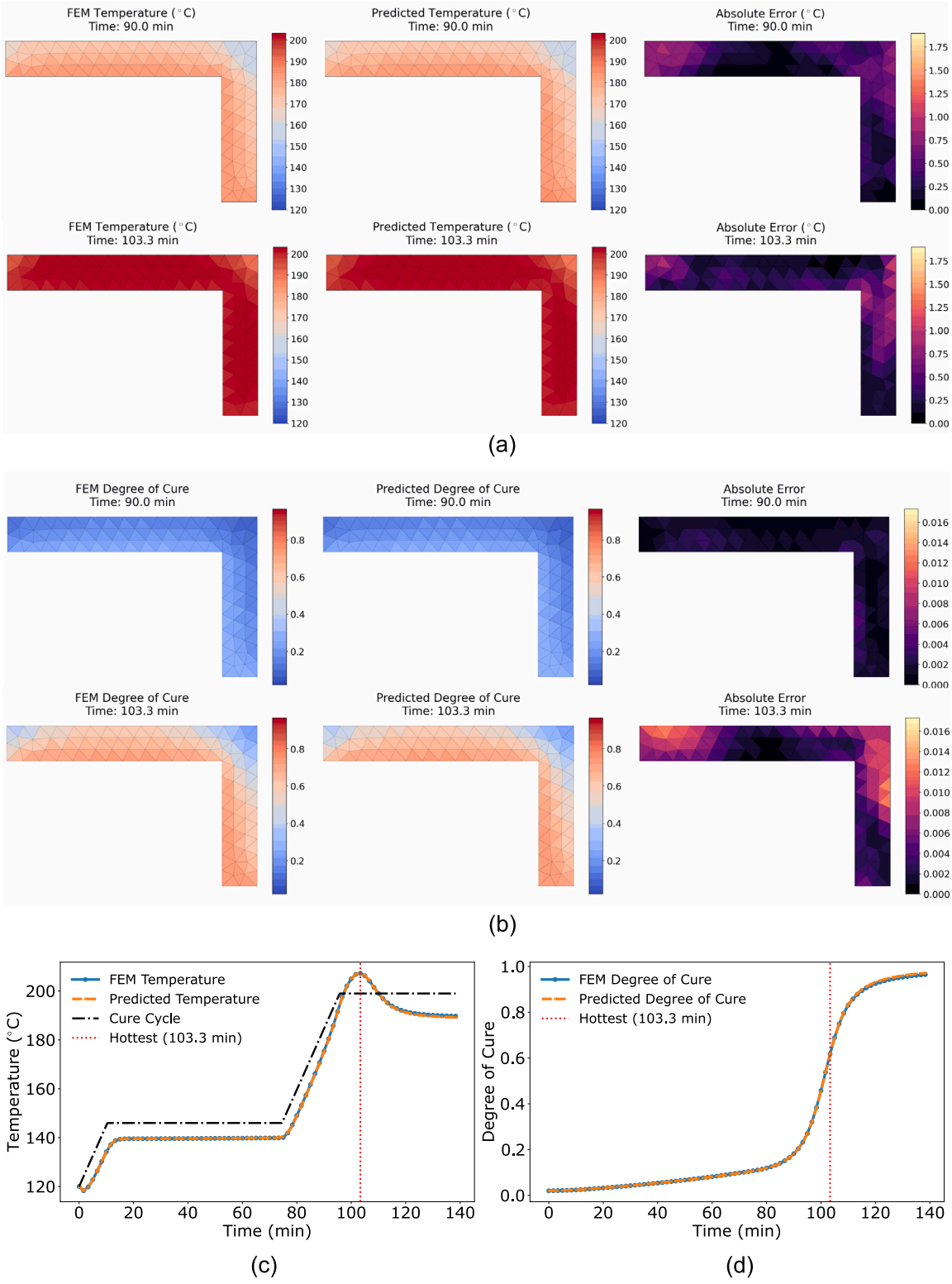



**Fig. 6.** Model predictions vs FEM results for (a) temperature and (b) degree of cure for the best simulation case in L-shaped domain (c) Temperature evolution at hottest node (d) Cure evolution at the hottest node.

geometries and materials are similar between different generations and such a model could be trained on the data of previous generations and be used in the early-stage process design of the next generation. Such an approach would nullify the impact of training dataset generation, and can also be potentially merged with current research on physics-informed networks [20] that reduce the need for large amounts of high-fidelity training data, further reducing initial computational effort and improving the extrapolation performance.

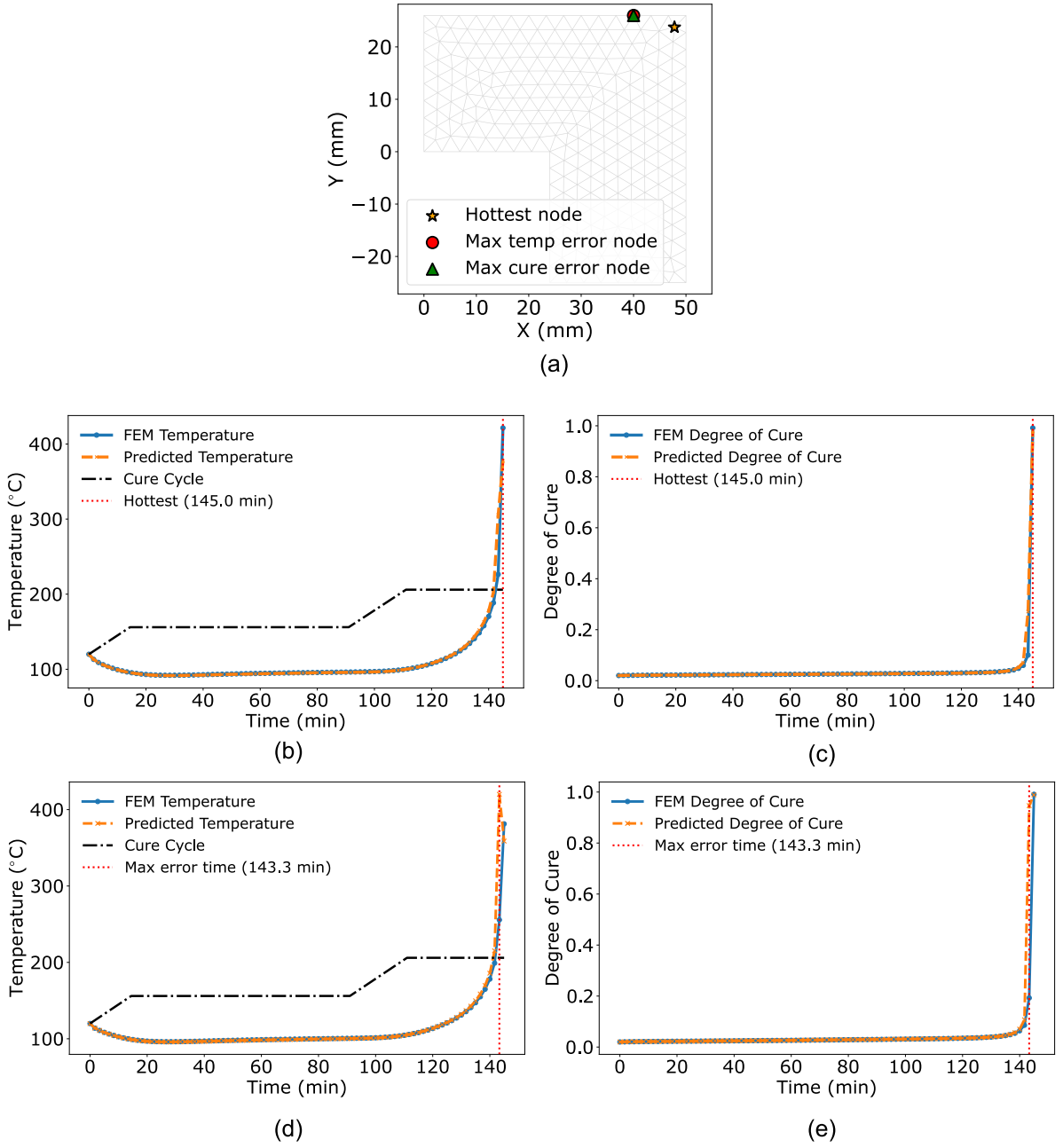



**Fig. 7.** (a) Location of hottest and maximum error nodes in the worst case prediction for L-shaped domain with 26 mm thickness (b) Temperature and (c) cure evolution at the hottest node (d) Temperature and (e) cure evolution at the node with maximum error.

### *4.4. Generalisability of the model*

With the aim of inspecting the robustness of the approach to the introduction of anisotropic conductivity, a new model was trained on a dataset with simulations including longitudinal conductivity $k_{xx} = 6.45$ $W\,m^{-1}\,K^{-1}$, where $k_{xx} = k_{yy}$ are the longitudinal conductivities and $k_{zz} = 0.4$ $W\,m^{-1}\,K^{-1}$ is the transverse conductivity (thickness in the $z$-axis direction). This is representative of a quasi-isotropic layup, where the in-plane conductivity is uniform in all directions and conductivity in the through thickness direction, transverse to the fibre directions, remains low. Conductivity values were adopted from literature [22] and were considered constant throughout the curing process for simplicity. The conductivity values are homogenised considering rule of mixtures, where $k_{xx}$ is calculated considering the longitudinal conductivity of the fibres, while $k_{zz}$ considers the transverse conductivity of the fibres.

Figs. 10(a) (b) and (c) show the results for a simulation case on a 10 mm thick part, where the error is only 0.81 °C for temperature prediction and 0.011 for degree of cure prediction. Likewise, Figs. 10(d) (e) and (f) show the results for a 24 mm thick part, which is also the lowest accuracy prediction case in the test set. The maximum errors are 32.3 °C and 0.207 for temperature and degree of cure predictions.

Overall, this newly trained model had $R^2 = 0.9976$ and RMSE = 1.15 °C for temperature predictions across the test set. For degree of cure predictions, the model achieved $R^2 = 0.9995$ and RMSE = 0.0078.

Compared to the results of the previously trained model that considers isotropic conductivity, this new model with both transverse and longitudinal conductivities has improved $R^2$ and lower errors, indicating better performance. The introduction of anisotropy results in better in-plane heat dissipation, which results in less steep temperature and degree of cure rates for the model to learn. Consequently, this improves

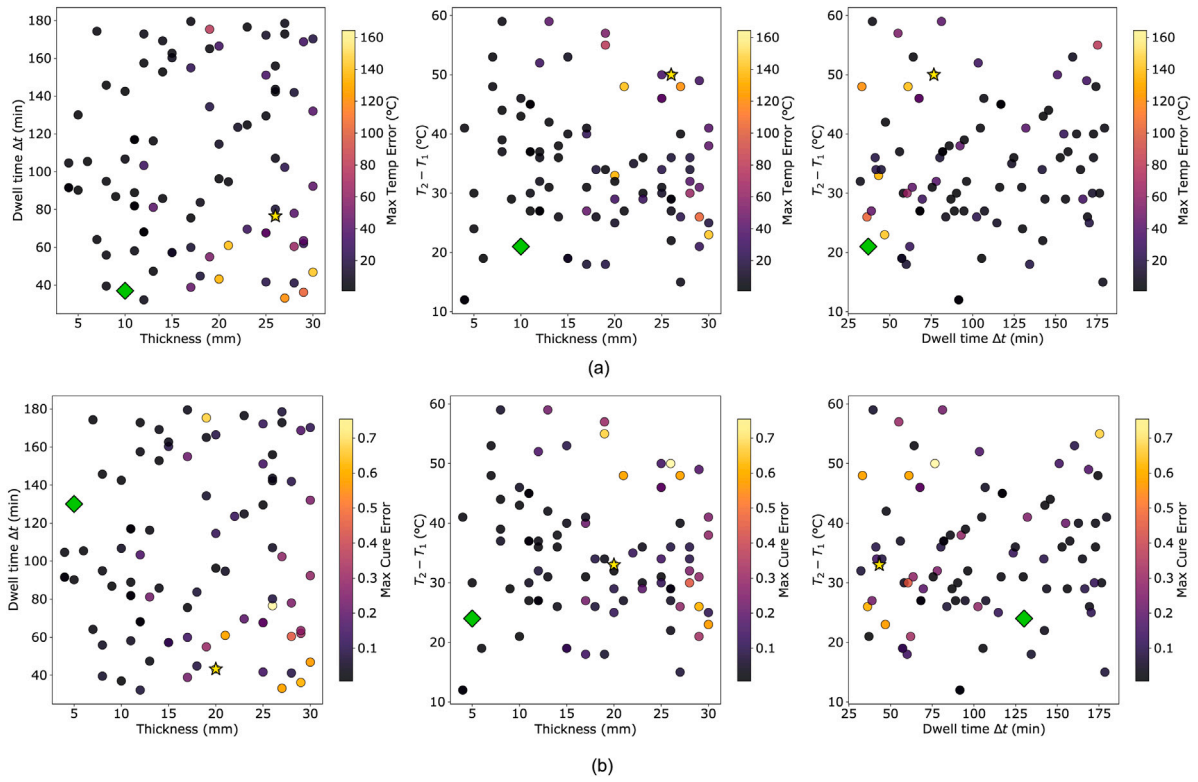



**Fig. 8.** Correlation of simulation parameters with maximum prediction errors for (a) Temperature and (b) Degree of Cure, with the best and worst prediction cases marked by a diamond and a star, respectively.

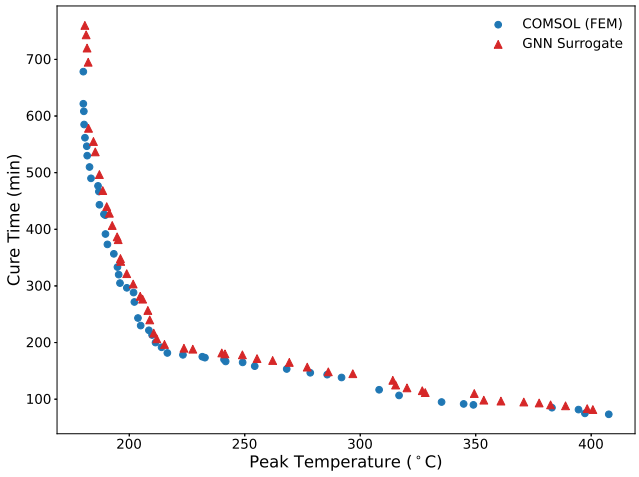



**Fig. 9.** Pareto fronts resulting from multi-objective cure cycle optimisation using the GNN surrogate represented by red triangles, and using FEM represented by blue circles.

the performance of the model and is an indicator of the potential of this approach to be robust to anisotropy in the material property. It is also reflected in the parity plot in Fig. 11 where the peak temperature does not cross 250 °C. The higher-density points are also more clustered around the diagonal compared to the parity plot in Fig. 5. This opens the door for further research on how the model performs when more complexities are introduced, such as accounting for the cure-dependent material properties.

With regard to generalisability across geometries, the previous model's prediction capability had been tested on T- and U- shapes, which were not part of the training data then. It tested the model's ability to extrapolate beyond the training set range. These results are presented in Appendix D. As expected, these results had greater error relative to FEM results. The physical trend in the evolution of temperature and degree of cure was captured, with a lag or lead in the predicted rates, more pronounced for the T- shape, resulting in the errors. This showed that the model had the potential to be more robust across different geometries if the training set had greater geometric variability.

The newly trained model, with both transverse and longitudinal conductivities, also had 256 cure simulations each for T and U-shaped domains, in addition to the flat and L-shaped cross sections. Fig. 12 shows this model's predictions for the T and U-shaped domains for a combination of simulation parameters previously unseen by the model during training. The hottest node is highlighted in red. The predictions of the model agree well with FEM results. The maximum errors in temperature predictions are 1.4 °C and 1.3 °C for these T and U domains, respectively, while the maximum errors in the cure degree predictions are 0.012 and 0.004. These results indicate that the extent of the model's generalisability to geometries depends on the geometric variability in the training data distribution.

To test how transferable the model might be to another manufacturing process, it was used for simulating a case representative of resin transfer moulding (RTM). The boundary conditions were changed from having convection on one side to prescribed temperature boundary conditions on both sides, which assumes the part is enclosed in a heated mould. Fig. 13 shows these predictions at the hottest nodes for rectangular and L-shaped domains, both 15 mm thick.

The results do not compare well with FEM solutions. Despite the type of boundary node being encapsulated in the node feature vector, it appears the model struggles when the combination of boundary node types is different from what it sees during training. This suggests that just encoding the boundary node type is not enough to make the model transferable to other combination of boundary conditions, and in turn to other manufacturing processes. Incorporating a range of boundary

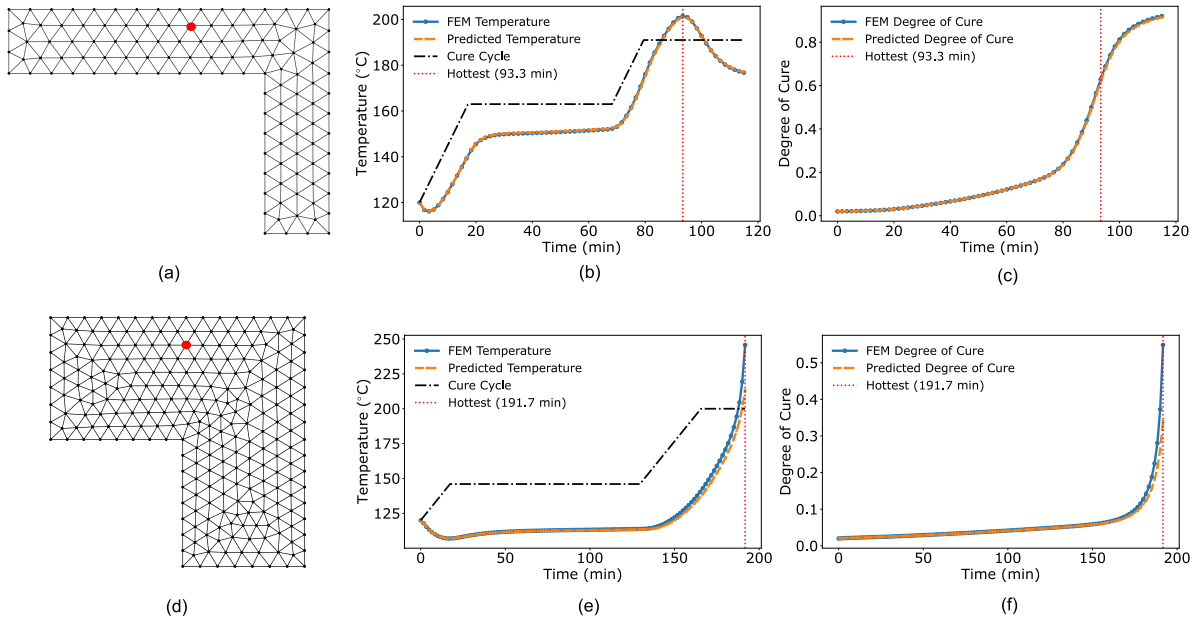



**Fig. 10.** Results of the trained model including transverse and longitudinal conductivities for a good case (a) and worst case (d), showing the (b & e) temperature and (c & f) degree of cure at the hottest node (highlighted in red)).

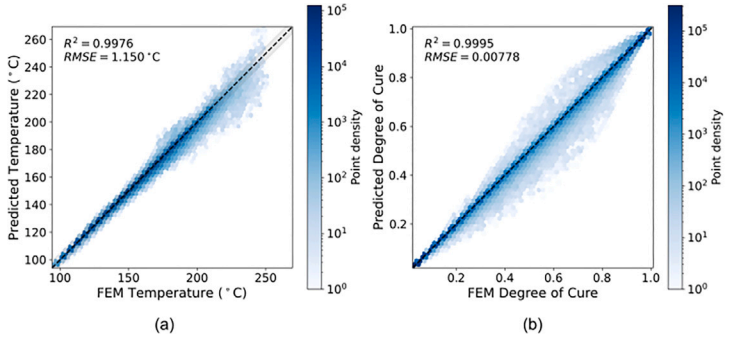



**Fig. 11.** Parity plots of (a) Temperature and (b) Degree of cure predictions for the model trained on data with longitudinal and transverse conductivities.

conditions representative of different processes, and including them as nodal or global feature vectors in the graphs is a potential way ahead for making a single model that can be applicable to different processes. This is envisioned as high-priority among considerations for further work.

## 5. Conclusions

This work investigated the suitability of a MeshGraphNet-inspired Graph Neural Network architecture as a surrogate model for thermo-chemical cure simulations of a composite material with a thermosetting matrix. The study focused on adapting the encode-process-decode architecture and assessing its predictive accuracy. The approach was trained and evaluated using two dimensional finite element simulations of composite curing subjected to cure cycles with two dwell stages.

The results showed that the model can reproduce predictions of nodal evolution of temperature and cure degree with near FEM accuracy within the training distribution, with single-step prediction times in the order of milliseconds. The model effectively captured the through-thickness thermal response, including the timing and magnitude of the exothermic temperature peaks.

However, for combinations of aggressive cure cycles and high thicknesses, where the peak temperatures exceeded 250 °C, the model struggled to predict accurately. In these cases with the highest errors, it was observed that the physical trend in the temperature and cure histories was captured with a time shift, thus causing high errors when comparing instantaneous nodal predictions with FEM. The model would benefit from additional data from these regimes. This is a limitation of the approach of sampling some cure cycle parameters to avoid generating hundreds of thousands of FEM simulations in the training data. This study refrained from generating more data to improve the model's performance further, because it was deemed unfruitful to consider cases where the temperatures are much higher than the material's degradation temperature, and would not be useful in a realistic manufacturing case.

Regarding generalisation across geometries using graphs, the approach struggles in predicting accurately when extrapolating to shapes outside the training distribution. When a new model was trained explicitly on additional data with T and U-shaped samples, it accurately predicted temperature and cure history for these shapes. It motivates building a more robust model for further work by increasing the geometric variability in the training set, which is a step towards developing a single model applicable to different parts.

To test the potential speed advantage of this cure modelling approach, the GNN was utilised in a multi-objective optimisation exercise to optimise the cure cycle for a 20 mm thick part. It was used as an FEM solver replacer in a framework that used a genetic algorithm to minimise peak temperature as well as total curing time. The Pareto front recovered from using the GNN surrogate closely matched that

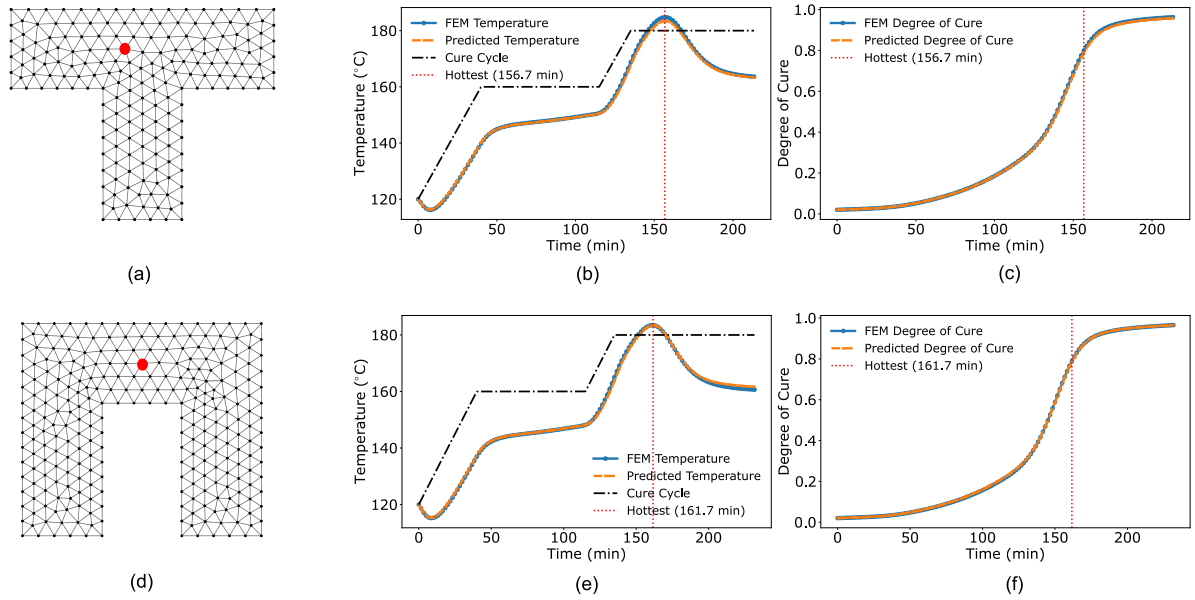



**Fig. 12.** Results of model trained with T and U shapes in the training set, with the hottest node highlighted in red (a) T-shaped domain with 15 mm thickness (b) Predicted temperature evolution at hottest node (c) Predicted degree of cure at hottest node (d) U-shaped domain with 15 mm thickness (e) Predicted temperature evolution at hottest node (f) Predicted degree of cure at hottest node.

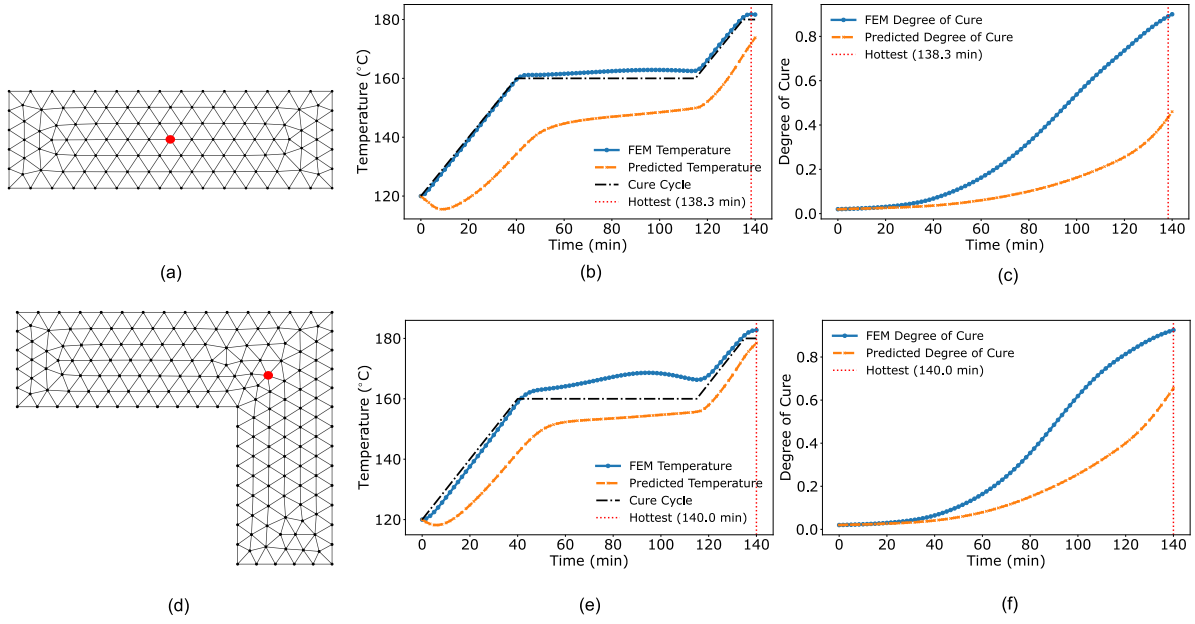



**Fig. 13.** Results of model predictions on (a) rectangular and (d) L-shaped domains showing evolution of (b & e) temperature and (c & f) degree of cure at the hottest node which is highlighted in red.

of the COMSOL finite-element simulation-based optimisation, with a runtime reduction of about 96.3%. It shows the potential of such an approach in early-stage process design.

Simplifications had been made to the thermo-chemical simulations for the purpose of this initial, exploratory study. The results of the model when including both longitudinal and transverse material conductivities are highly accurate, motivating further enhancement with inclusion of cure-dependent properties. Future development will also involve a thermo-chemo-mechanical model, facilitating prediction of cure shrinkage, deformation, and residual stresses, with the aim of developing a workflow for a full manufacturing scenario.

From a perspective of applicability to other manufacturing processes, it was seen that the model did not perform well when changing boundary conditions to represent an RTM process instead of the VARTM conditions it was trained on. It motivates the need for further research on inclusion of a range of boundary condition parameters in training as well in the graph construction methodology, to make the model adaptable to different processes.

Overall, the present study shows that GNN-based surrogate modelling is a promising route for fast cure simulation of thermoset composites, which can accelerate early-stage process design. It is a step towards developing more general and efficient models for applicability in real manufacturing scenarios.

## CRediT authorship contribution statement

**Ashish V. Hegde:** Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Dimitrios Zarouchas:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Baris Caglar:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ashish Hegde reports financial support was provided by Ministry of Economic Affairs and Climate Policy. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The project is made possible in part by a contribution from the National Growth Fund program NXTGEN Hightech Composites 01: Digital composite automation for sustainable aviation (The Netherlands). The authors acknowledge the use of computational resources of the DelftBlue supercomputer, provided by Delft High Performance Computing Centre [37].

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compositesa.2026.110186.

## Data availability

The data used in this study is freely accessible on Zenodo at https://doi.org/10.5281/zenodo.20177475. The trained model and source code are also open source and accessible on GitHub - https://github.com/camp-lab-tud/GNN-cure-simulation. This is intended to promote the use of machine learning in the composites field. Readers are encouraged to explore the code and adapt it as required.

## References

[1] Mallick PK. Processing of polymer matrix composites. Boca Raton London New York: CRC Press; 2018, http://dx.doi.org/10.1201/9781315157252.

[2] Wisnom MR, Gigliotti M, Ersoy N, Campbell M, Potter KD. Mechanisms generating residual stresses and distortion during manufacture of polymer–matrix composite structures. In: Internal stresses in polymer composites, Compos Part A: Appl Sci Manuf In: Internal stresses in polymer composites, 2006;37(4):522–9. http://dx.doi.org/10.1016/j.compositesa.2005.05.019.URL https://www.sciencedirect.com/science/article/pii/S1359835X0500179X,

[3] Antonucci V, Giordano M, Hsiao K-T, Advani SG. A methodology to reduce thermal gradients due to the exothermic reactions in composites processing. Int J Heat Mass Transfer 2002;45(8):1675–84. http://dx.doi.org/10.1016/S0017-9310(01)00266-6, URL https://www.sciencedirect.com/science/article/pii/S0017931001002666.

[4] Wang B, Fan S, Chen J, Yang W, Liu W, Li Y. A review on prediction and control of curing process-induced deformation of continuous fiber-reinforced thermosetting composite structures. Compos Part A: Appl Sci Manuf 2023;165:107321. http://dx.doi.org/10.1016/j.compositesa.2022.107321, URL https://www.sciencedirect.com/science/article/pii/S1359835X22005024.

[5] Anandan S, Dhaliwal GS, Huo Z, Chandrashekhara K, Apetre N, Iyyer N. Curing of thick thermoset composite laminates: Multiphysics modeling and experiments. Appl Compos Mater 2018;25(5):1155–68. http://dx.doi.org/10.1007/s10443-017-9658-9, URL http://link.springer.com/10.1007/s10443-017-9658-9.

[6] Larson MG, Bengzon F. The finite element method: Theory, implementation, and applications. Texts in computational science and engineering, (10). Berlin Heidelberg: Springer; 2013, http://dx.doi.org/10.1007/978-3-642-33287-6.

[7] Ruiz E, Trochu F. Numerical analysis of cure temperature and internal stresses in thin and thick RTM parts. Compos Part A: Appl Sci Manuf 2005;36(6):806–26. http://dx.doi.org/10.1016/j.compositesa.2004.10.021, URL https://linkinghub.elsevier.com/retrieve/pii/S1359835X0400260X.

[8] Goodfellow I, Bengio Y, Courville A. Deep learning. Adaptive computation and machine learning, Cambridge, Mass: The MIT Press; 2016.

[9] Caglar B, Broggi G, Ali MA, Orgéas L, Michaud V. Deep learning accelerated prediction of the permeability of fibrous microstructures. Compos Part A: Appl Sci Manuf 2022;158:106973. http://dx.doi.org/10.1016/j.compositesa.2022.106973, URL https://www.sciencedirect.com/science/article/pii/S1359835X22001622.

[10] Jean JG, Broggi G, Caglar B. An image-based deep learning framework for flow field prediction in arbitrary-sized fibrous microstructures. Compos Part A: Appl Sci Manuf 2026;200:109337. http://dx.doi.org/10.1016/j.compositesa.2025.109337, URL https://www.sciencedirect.com/science/article/pii/S1359835X25006311.

[11] Cassola S, Duhovic M, Schmidt T, May D. Machine learning for polymer composites process simulation – a review. Compos Part B: Eng 2022;246:110208. http://dx.doi.org/10.1016/j.compositesb.2022.110208, URL https://www.sciencedirect.com/science/article/pii/S1359836822005820.

[12] Hui X, Xu Y, Zhang W, Zhang W. Cure process evaluation of CFRP composites via neural network: From cure kinetics to thermochemical coupling. Compos Struct 2022;288:115341. http://dx.doi.org/10.1016/j.compstruct.2022.115341, URL https://www.sciencedirect.com/science/article/pii/S0263822322001465.

[13] Carlone P, Aleksendrić D, Ćirović V, Palazzo GS. Meta-modeling of the curing process of thermoset matrix composites by means of a FEM–ANN approach. Compos Part B: Eng 2014;67:441–8. http://dx.doi.org/10.1016/j.compositesb.2014.08.022, URL https://www.sciencedirect.com/science/article/pii/S1359836814003333.

[14] Zhang F, Guo H, Wang W, Du J, Liu Y, Wang H. Fast prediction of curing residual stress in polymer-matrix composites by a deep learning method. Polym Compos 2025;46(9):8424–41.

[15] Humfeld KD, Gu D, Butler GA, Nelson K, Zobeiry N. A machine learning framework for real-time inverse modeling and multi-objective process optimization of composites for active manufacturing control. Compos Part B: Eng 2021;223:109150. http://dx.doi.org/10.1016/j.compositesb.2021.109150, URL https://www.sciencedirect.com/science/article/pii/S135983682100531X.

[16] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707. http://dx.doi.org/10.1016/j.jcp.2018.10.045, URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[17] Amini Niaki S, Haghighat E, Campbell T, Poursartip A, Vaziri R. Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture. Comput Methods Appl Mech Engrg 2021;384:113959. http://dx.doi.org/10.1016/j.cma.2021.113959, URL https://www.sciencedirect.com/science/article/pii/S0045782521002966.

[18] Hamilton WL, Ying R, Leskovec J. Representation learning on graphs: Methods and applications. 2018, http://dx.doi.org/10.48550/arXiv.1709.05584, URL http://arxiv.org/abs/1709.05584. arXiv:1709.05584[cs].

[19] Pfaff T, Fortunato M, Sanchez-Gonzalez A, Battaglia PW. Learning mesh-based simulation with graph networks. 2021, arXiv:2010.03409[cs]. URL http://arxiv.org/abs/2010.03409.

[20] Würth T, Freymuth N, Zimmerling C, Neumann G, Kärger L. Physics-informed MeshGraphNets (PI-MGNs): Neural finite element solvers for non-stationary and nonlinear simulations on arbitrary meshes. Comput Methods Appl Mech Engrg 2024;429:117102. http://dx.doi.org/10.1016/j.cma.2024.117102, URL https://www.sciencedirect.com/science/article/pii/S004578252400358X.

[21] Yousefi A, Lafleur PG, Gauvin R. Kinetic studies of thermoset cure reactions: A review. Polym Compos 1997;18(2):157–68. http://dx.doi.org/10.1002/pc.10270, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pc.10270. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pc.10270.

[22] Struzziero G, Skordos AA. Multi-objective optimisation of the cure of thick components. Compos Part A: Appl Sci Manuf 2017;93:126–36. http://dx.doi.org/10.1016/j.compositesa.2016.11.014, URL https://www.sciencedirect.com/science/article/pii/S1359835X16303955.

[23] Brauner C, Soprano P, Herrmann AS, Meiners D. Cure-dependent thermochemical modelling and analysis of the manufacturing process of an aircraft composite frame. J Compos Mater 2015;49(8):921–38. http://dx.doi.org/10.1177/0021998314527777, Publisher: SAGE Publications Ltd STM.

[24] Hsiao K-T, Heider D. 10 - Vacuum assisted resin transfer molding (VARTM) in polymer matrix composites. In: Advani SG, Hsiao K-T, editors. Manufacturing techniques for polymer matrix composites (pmcs). Woodhead publishing series in composites science and engineering, Woodhead Publishing; 2012, p. 310–47. http://dx.doi.org/10.1533/9780857096258.3.310, URL https://www.sciencedirect.com/science/article/pii/B9780857090676500109.

[25] Struzziero G, Teuwen JJE. A fully coupled thermo-mechanical analysis for the minimisation of spring-in and process time in ultra-thick components for wind turbine blades. Compos Part A: Appl Sci Manuf 2020;139:106105. http://dx.doi.org/10.1016/j.compositesa.2020.106105, URL https://www.sciencedirect.com/science/article/pii/S1359835X20303444.

[26] Struzziero G, Teuwen JJE, Skordos AA. Numerical optimisation of thermoset composites manufacturing processes: A review. Compos Part A: Appl Sci Manuf 2019;124:105499. http://dx.doi.org/10.1016/j.compositesa.2019.105499, URL https://www.sciencedirect.com/science/article/pii/S1359835X19302489.

[27] Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R, Gulcehre C, Song F, Ballard A, Gilmer J, Dahl G, Vaswani A, Allen K, Nash C, Langston V, Dyer C, Heess N, Wierstra D, Kohli P, Botvinick M, Vinyals O, Li Y, Pascanu R. Relational inductive biases, deep learning, and graph networks. 2018, http://dx.doi.org/10.48550/arXiv.1806.01261, arXiv:1806.01261. URL http://arxiv.org/abs/1806.01261.

[28] Fey M, Lenssen JE. Fast graph representation learning with PyTorch geometric. 2019, http://dx.doi.org/10.48550/arXiv.1903.02428, arXiv:1903.02428[cs] Citaton Key:pyg. URL http://arxiv.org/abs/1903.02428.

[29] Elfwing S, Uchibe E, Doya K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. In: Special issue on deep reinforcement learning, Neural Netw In: Special issue on deep reinforcement learning, 2018;107:3–11. http://dx.doi.org/10.1016/j.neunet.2017.12.012.URL https://www.sciencedirect.com/science/article/pii/S0893608017302976,

[30] Ba JL, Kiros JR, Hinton GE. Layer normalization. 2016, http://dx.doi.org/10.48550/arXiv.1607.06450, arXiv:1607.06450[stat]. URL http://arxiv.org/abs/1607.06450.

[31] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97. http://dx.doi.org/10.1109/4235.996017, URL https://ieeexplore.ieee.org/document/996017.

[32] Blank J, Deb K. Pymoo: Multi-objective optimization in Python. IEEE Access 2020;8:89497–509. http://dx.doi.org/10.1109/ACCESS.2020.2990567, URL https://ieeexplore.ieee.org/document/9078759.

[33] Hennig J, Maeder A, AnnikaSchultheiss, Feder J. MPh-py/MPh: 1.3.0. 2025, http://dx.doi.org/10.5281/zenodo.17555089.

[34] Formicola C, De Fenzo A, Zarrelli M, Frache A, Giordano M, Camino G, et al. Synergistic effects of zinc borate and aluminium trihydroxide on flammability behaviour of aerospace epoxy system. Express Polym Lett 2009;3(6):376–84.

[35] Kaddour J, Key O, Nawrot P, Minervini P, Kusner MJ. No train no gain: Revisiting efficient training algorithms for transformer-based language models. Adv Neural Inf Process Syst 2023;36:25793–818.

[36] Dehghani M, Tay Y, Arnab A, Beyer L, Vaswani A. The efficiency misnomer. In: International conference on learning representations. 2022, URL https://openreview.net/forum?id=iulEMLYh1uR.

[37] Delft High Performance Computing Centre (DHPC). DelftBlue supercomputer (phase 2). 2024, https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.