Key Engineering Materials
ISSN: 1662-9795, Vol. 1050, pp 31-42
doi:10.4028/p-kaH2TN

Submitted: 2025-12-19
Revised: 2026-02-18
Accepted: 2026-02-18
Online: 2026-04-16

Algebraic Hierarchical Graph Neural Networks for Forming Simulation of Thermoplastic Composite Materials

Tobias Würth[1,a*], Niklas Freymuth[2,b], Johannes Mitsch[1,c], Philipp Dahlinger[2,d], Tai Hoang[2,e], Gerhard Neumann[2,f] and Luise Kärger[1,g]

[1]Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Karlsruhe, Germany

[2]Autonomous Learning Robots, Karlsruhe Institute of Technology, Karlsruhe, Germany

[a]tobias.wuerth@kit.edu, [b]niklas.freymuth@kit.edu, [c]Johannes.Mitsch@kit.edu, [d]philipp.dahlinger@kit.edu, [e]tai.hoang@kit.edu, [f]gerhard.neumann@kit.edu, [g]luise.kaerger@kit.edu

Keywords: Forming, Graph Network Simulators, Algebraic Multi-Grid Coarsening, Data-Driven Simulation

Abstract. Stamp forming of fiber-reinforced thermoplastic composite materials is governed by large deformations, anisotropic and rate-dependent material behavior, and frictional multi-body contact, making high-fidelity finite element simulations expensive and often impractical for rapid design studies and process optimization. We leverage recent advancements in Machine Learning-based simulations and tailor Algebraic-hierarchical Message Passing Networks (AMPNs) to stamp forming simulation of composite materials. To efficiently handle multi body contact during forming, we model the laminates by a multi-layer graph with explicit ply–ply and tool–ply contact and extend AMPNs by local component-wise contact edges. Using a multiscale graph hierarchy, the method captures local wrinkling effects, global material draw-in, and contact-driven deformation across the full laminate. Trained on high-fidelity data from state-of-the-art Finite Element Method (FEM) simulations, the surrogate accurately simulates the stamp forming process for unseen process settings, while reducing simulation times from hours to seconds, enabling approximately real time simulation of large, complex geometries.

Introduction

The performance of fiber-reinforced composite components depends heavily on stamp forming processes that shape flat laminates into complex geometries [1]. These materials are lightweight while offering high specific strength and fatigue resistance, making them increasingly attractive for the automotive and aerospace industries [2, 3].

Simulation of the forming process enables a virtual process optimization and helps evaluate design decisions, providing deeper insights into complex systems without requiring expensive real-world experiments. Such simulations model the physical system using Partial Differential Equations (PDEs), which are usually approximated by numerical discretizations like the FEM [4]. A critical challenge in simulating the stamp forming process of fiber reinforced composite materials is the prediction of manufacturing defects such as wrinkling, i.e., high-frequency, out-of-plane buckling modes. Accurately capturing such defects requires fine-grained meshes, narrow numerical time-integration schemes [2] and advanced simulation models [5], which quickly results in high computational cost that makes downstream applications infeasible [6].

Recently, research in Machine Learning (ML)-based models has aimed to speedup simulations by developing neural surrogates that can approximate complex physical dynamics [7, 8, 9]. Physics-Informed Neural Networks directly minimize PDE residuals, using Multilayer Perceptrons (MLPs) [10, 11], Convolutional Neural Networks (CNNs) [12, 13] or Graph Neural Networks (GNNs) [14, 15]. Neural Operators learn mesh-resolution-independent solution mappings [16, 17]. Complementary to theses approaches, data-driven Graph Network Simulator (GNS) have been shown to be particularly effective for modeling mesh-based dynamics [8, 9, 18]. Recent extensions consider diffusion-based

inference to improve accuracy for image-based surrogates [19, 20] as well as for mesh-based simulations [21] and meta-learning to improve generalization to unseen scenarios [22, 23, 24]. These models mimic simulation data from numerical solvers by using their own predictions to estimate the state residuals between successive time steps [17]. The seminal MeshGraphNet (MGN) [8] encodes states using local connectivity, but struggles to scale to large domains due to a limited receptive field. As a remedy, recent work considers global attention [25] and, increasingly, hierarchical mesh representations [26, 21]. Such globally receptive GNSs are fully differentiable and orders of magnitude faster than classical simulators, which unlocks applications in engineering design and manufacturing optimization [27, 28, 11].

In this work, we leverage these modern advances in GNS to establish a fast and accurate surrogate model for forming process of thermoplastic composite materials. We build upon the Rolling Diffusion-Batched Inference Network (ROBIN) framework [21] and utilize Algebraic-hierarchical Message Passing Networks (AMPNs) to process mesh representations based on Algebraic multigrid (AMG) coarsening [29, 30]. Crucially, we depart from diffusion-based inference of ROBIN [21], and instead use a simple next-step predictor based on AMPNs at inference time. In our stamp forming setting, we demonstrate that this choice improves accuracy and reduces inference cost compared to diffusion-based sampling. We further propose a sparse connectivity strategy for the inter-ply graph construction, drastically reducing the number of required edges for multi-layer simulations without sacrificing the simulation fidelity of local interactions.

We validate the presented approach on a stamp forming task involving a multi-directional stacking sequence $[0°/45°/90°/-45°]$, which promotes the local formation of wrinkles due to high resistance against shear deformation. Figure 1 visualizes the problem setup. Our experiments demonstrate that the proposed method achieves near-FEM accuracy in capturing local wrinkling phenomena, while reducing computational cost by orders of magnitude compared to explicit solvers.

Thermoforming Simulation

Simulation model. In order to apply AMPNs to the thermoforming process, we generate simulation data using a high-fidelity FEM simulation model. The simulation model is implemented in the commercial FEM software Abaqus/Explicit and is based on the work of Dörr et al. [31, 32], who developed an isothermal, layer-by-layer approach for thermoforming simulation for thermoplastic composite materials. The model accounts for the material behavior of the individual plies and their inter-ply interactions and provides essential features for accurate thermoforming simulations, including tool-ply and ply-ply contact modeling [32], decoupled membrane-bending behavior, and rate-dependent material modeling of both membrane and bending deformation mechanisms [31]. The model has been experimentally validated and accurately captures the deformation behavior during forming into complex geometries for various stacking sequences, including material draw-in, wrinkle formation, and the resulting fiber orientation.

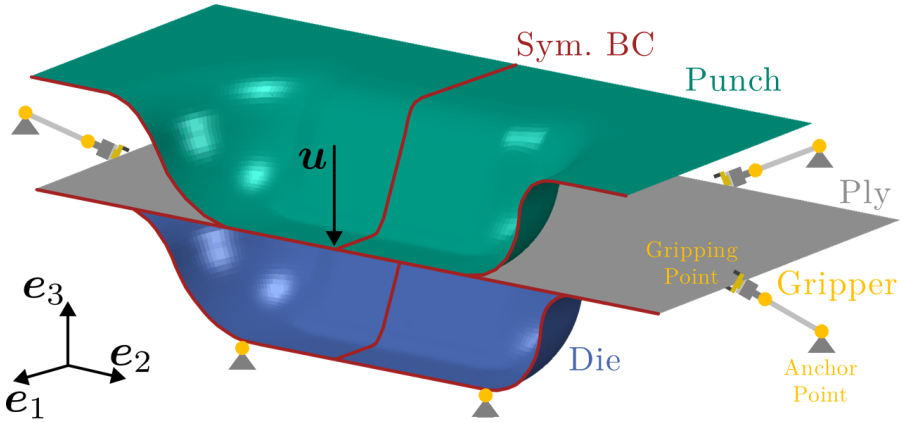

Fig. 1: Numerical test setup for the forming simulation of a double-domed geometry. A junglegreen-punch geometry presses multiple plies against a stationary die. The plies are held by multiple grippers.

To account for the influence of tensile grippers on the deformation behavior of the thermoplastic blank during forming, the methodology proposed by Poppe et al. [33] is adopted. This approach involves defining a space-fixed anchor point on the gripper frame for the tensile gripper, as well as a blank-fixed gripping point where the material is clamped by a needle. Using a one-dimensional translator element, the gripper force is applied as a concentrated load at the blank-fixed gripping point. This load acts along the line connecting the anchor and gripping point, ensuring that the force is always directed toward the space-fixed anchor. The combination of the translator element and a hinge element enables the gripping point to rotate about the needle, allowing nodding motions of the gripper during forming.

Dataset. We use this simulation model to generate representative training data for the AMPNs. Fig. 1 illustrates the numerical thermoforming model implemented in Abaqus/Explicit. The double-domed geometry considered in this study is commonly used in the forming literature [34, 35, 36], as its pronounced double curvature induces strongly heterogeneous deformation and can trigger deformation-induced wrinkling, yielding geometrically complex, out-of-plane ply shapes. We implement the forming process in a displacement-controlled manner by prescribing a Dirichlet boundary condition ($\mathbf{u} = -60$ mm $\mathbf{e}_3$) on the punch geometry (green) over a time period of $T{=}1$ s. We consider a laminate of carbon fibers in a PA6 thermoplastic matrix consisting of four plies with orientations $[0°/45°/90°/-45°]$ relative to the $\mathbf{e}_1$-axis (cf. Fig. 1). Eight tensile grippers are distributed along the outer edges of the blank to apply gripper forces during forming. We exploit the two-fold symmetry of the tool geometry, and only simulate one quarter of the model to improve computational efficiency. We discretize the blank using $3\,200$ triangular membrane and shell elements per ply, while we discretize the tool with $4\,258$ and $5\,100$ quadrilateral elements for the stamp and die, respectively. We model contact between the blank and the tools using the general contact algorithm in Abaqus/Explicit. In total, we conduct 684 simulations with varying gripper positions along the blank edges to generate representative training data for the AMPNs, while we use 147 simulations for validation and testing. For each simulation trial, we sample the position of both grippers independently and uniformly along the corresponding blank edges. In addition, we sample the magnitude of the pulling force applied by each gripper independently and uniformly within the force limits of the employed gripper, from $0\,\mathrm{N}$, where gripper-induced force acts on the plies, up to $100\,\mathrm{N}$. We discretize each simulation into 50 time steps, capturing the transient deformation behavior of the blank during the forming process. For each node of the finite element mesh, the resulting dataset provides nodal displacement vectors as well as the initial fiber orientation. These quantities, together with the gripper positions and pulling forces, constitute the input features used for training the AMPNs model. We executed the simulations on an Intel Xeon Gold 6230 processor, requiring approximately 275 hours of total computation time.

Methodology - Algebraic-Hierarchical Message Passing Networks (AMPNs)

To make thermoforming simulation practical for iterative engineering workflows, we learn a surrogate that reproduces the FEM trajectories at a fraction of the runtime. We build upon AMPNs [21] because their hierarchical graph representation supports the multiscale nature of stamp forming by capturing local contact interactions and wrinkle formation while also propagating global deformation and draw-in effects across the full laminate.

The AMPNs predict the residual displacement $\Delta\tilde{\mathbf{u}}^t \approx \Delta\mathbf{u}^t = \mathbf{u}^t - \mathbf{u}^{t-1}$ in stamp forming of the current time step t given the displacement $\mathbf{u}^{t-1}$ from the previous time step $t-1$. During inference, we rollout the AMPN autoregressivly by updating the displacements with $\mathbf{u}^t \approx \mathbf{u}^{t-1} + \Delta\tilde{\mathbf{u}}^t$ and use that state as new input for predicting the next residual $\Delta\mathbf{u}^{t+1}$. We build up on ROBIN [21] and implement the AMPNs as a special case of ROBIN with a single diffusion step. In this setting, ROBIN directly predicts the residual update $\Delta\mathbf{u}^t$, and the training objective reduces to a conventional Mean Squared Error (MSE) loss against the ground-truth FEM residual update. AMPNs operate on a hierarchy of graphs with different resolutions, where coarse hierarchy levels allow global deformation predictions, while finer levels allow for refining fine details. Given a simulation mesh, we create a fine, initial graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{\mathrm{M}})$, where $\mathcal{V}$ denotes mesh nodes and $\mathcal{E}^{\mathrm{M}}$ mesh edges. From this mesh, we create a hierarchical graph $\mathcal{G}^{\mathrm{H}} = (\mathcal{V}^{0:L}, \mathcal{E}^{0:L,\mathrm{M}} \cup \mathcal{E}^{0:L,\mathrm{C}} \cup \mathcal{E}^{0:L-1,\mathrm{D}} \cup \mathcal{E}^{0:L-1,\mathrm{U}})$ based on root-node smoothed aggregation [30], using the Algebraic multigrid (AMG) solver implemented in `PyAMG` [37]. Given the adjacency matrix A^0 of the fine simulation mesh, we use the AMG solver to create a hierarchy of graph nodes $\mathcal{V}^{0:L}$, mesh edges $\mathcal{E}^{0:L,\mathrm{M}}$, down- $\mathcal{E}^{0:L-1,\mathrm{D}}$ and upsampling edges $\mathcal{E}^{0:L-1,\mathrm{U}}$, where the superscript $0:L$ denotes the set of hierarchy levels $l \in \{0, \dots, L\}$ (with $l=0$ the finest level and $l=L$ the coarsest). In the hierarchy, the fine level graph $l=0$ corresponds to given fine simulation mesh of the FEM. For the considered double-domed geometry the AMG algorithm yields $L=3$, i.e., 3 coarse levels in addition to the fine level at $l=0$.

To learn contact, we follow previous works [8, 21] and connect nodes with contact edges $\mathcal{E}^{0:L,\mathrm{C}}$ by spatial proximity. However, conventional radius-based approaches [8] connect each node to all neighbors within a fixed radius, typically including nodes on the same component, leading to an unbounded number of edges in contact-rich stamp forming. Choosing a large radius anticipates upcoming contacts but can cause a prohibitive growth in edges and memory, whereas choosing a small radius misses imminent contacts unless very small time steps are used. We therefore connect each node to neighboring nodes of other components that are within a radius of 3 mm, but limit the number of connections to the 3 closest connections per node and component. That means that, in our stamp-forming setup with 4 plies, 1 stamp, and 1 die, each node can connect to at most 15 nodes from the other components. In contrast to static contact edges as in Würth et al. [21], we use time-dependent contact edges, i.e., we update the contact edges based on the current ply and tool locations at each time step, based on the current position.

Feature and encoder. For the fine mesh nodes (level $l=0$), the node features are comprised of a one-hot encoding n_i of the node type, the last predicted displacement residual $\Delta\mathbf{u}_i^{t-1}$, the prescribed tool motion $\Delta\mathbf{u}_{i,\mathrm{BC}}^t$ at the current time step t, the fiber orientation in the ply, and a component identifier. We add additional node features to inform the model about the gripper forces. Specifically, for each node position $\mathbf{x}_i$ and gripper position $\mathbf{g}_j$ we compute the distance vector $\mathbf{d}_{ij} = \mathbf{x}_i - \mathbf{g}_j$, its length $d_{ij} = \|\mathbf{d}_{ij}\|_2$, and the normalized direction $\hat{\mathbf{d}}_{ij} = \frac{\mathbf{d}_{ij}}{d_{ij}+\epsilon}$. We further include features based on an exponentially decaying distance weighting. For each node i, we compute the distance-weighted projected force $\mathbf{f}_i^{\text{proj-weighted}} = \sum_j \frac{\mathbf{f}_{ij}^{\text{directed}}}{\exp(\alpha \cdot d_{ij}+\epsilon)}$ with the distance projected gripper force $\mathbf{f}_{ij}^{\text{directed}} = (\mathbf{f}_j^{\text{gripper}} \cdot \hat{\mathbf{d}}_{ij}) \cdot \hat{\mathbf{d}}_{ij}$ as well as the distance-weighted gripper force field $\mathbf{f}_i^{\text{weighted}} = \sum_j \frac{\mathbf{f}_j^{\text{gripper}}}{\exp(\alpha \cdot d_{ij}+\epsilon)}$ and its norm $f_i^{\text{weighted}} = \|\mathbf{f}_i^{\text{weighted}}\|_2$. We set $\alpha = 3e{-}2$ and $\epsilon = 1e{-}8$. For the hierarchical nodes, i.e., for all nodes with $l>0$, we only use the one hot encoding n_i of the node type as a feature. As we follow the ROBIN setup, we additionally provide the model with a noisy input sample drawn from a standard Gaussian distribution

$\mathcal{N}(0,1)$. This additional input does not significantly impact the model's predictions, but allows us to directly compare to ROBIN's original diffusion-based predictions without further modifications.

As edge features, we use the relative current node distance $\mathbf{x}_{ij}^t = \mathbf{x}_i^t - \mathbf{x}_j^t$ and its norm $|\mathbf{x}_{ij}^t|$ at all hierarchy levels. For mesh, downsampling, and upsampling edges, we additionally include the initial relative distance $\mathbf{x}_{ij}^0 = \mathbf{x}_i^0 - \mathbf{x}_j^0$ and its norms $|\mathbf{x}_{ij}^0|$. We apply linear layers that project the corresponding input features into a latent space of dimension d, yielding node embeddings $\mathbf{k}_i^n$ for $i \in \mathcal{V}^l$, mesh edge embeddings $\mathbf{e}_{ij}^{\mathrm{M},n}$ for $(i,j) \in \mathcal{E}^{l,\mathrm{M}}$, contact edge embeddings $\mathbf{e}_{ij}^{\mathrm{C},n}$ for $(i,j) \in \mathcal{E}^{\mathrm{C}}$, downsampling edge embeddings $\mathbf{e}_{ij}^{\mathrm{D},n}$ for $(i,j) \in \mathcal{E}^{l,\mathrm{D}}$, and upsampling edge embeddings $\mathbf{e}_{ij}^{\mathrm{U},n}$ for $(i,j) \in \mathcal{E}^{l,\mathrm{U}}$ for each level l at the initial step, i.e., at $n{=}0$. All features are normalized using statistics computed on the training set to obtain zero mean and unit variance. Finally, we also apply a Fourier encoding to condition the model on the current node level l [38, 21].

Processor and decoder. ROBIN uses Intra-Level-Message Passing Stacks [21] to update node- and edge-level embeddings in the Pre-Processing, Post-Processing and Solving blocks. The intra-level update at message passing step n for a given hierarchy level l is defined by

$$\begin{aligned}
\mathbf{e}_{ij}^{\mathrm{C},n+1} &= W_{\theta,\mathcal{E}^{\mathrm{C}}}^n \, \mathbf{e}_{ij}^{\mathrm{C},n} + f_{\theta,\mathcal{E}^{\mathrm{C}}}^n(\mathbf{k}_i^n, \mathbf{k}_j^n, \mathbf{e}_{ij}^{\mathrm{C},n}) \,, \\
\mathbf{e}_{ij}^{\mathrm{M},n+1} &= W_{\theta,\mathcal{E}^{\mathrm{M}}}^n \, \mathbf{e}_{ij}^{\mathrm{M},n} + f_{\theta,\mathcal{E}^{\mathrm{M}}}^n(\mathbf{k}_i^n, \mathbf{k}_j^n, \mathbf{e}_{ij}^{\mathrm{M},n}) \,, \\
\mathbf{k}_i^{n+1} &= W_{\theta,\mathcal{V}}^n \, \mathbf{k}_i^n + f_{\theta,\mathcal{V}}^n(\mathbf{k}_i^n, \bigoplus_j \mathbf{e}_{ij}^{\mathrm{C},n+1}, \bigoplus_j \mathbf{e}_{ij}^{\mathrm{M},n+1}) \,.
\end{aligned} \tag{1}$$

For Up- and Downsampling, we use Inter-Level-Message Passing Stacks [21], which update the embeddings at each message passing step n as

$$\begin{aligned}
\mathbf{e}_{ij}^{n+1} &= W_{\theta,\mathcal{E}}^n \, \mathbf{e}_{ij}^n + f_{\theta,\mathcal{E}}^n(\mathbf{k}_i^{\mathrm{rec},n}, \mathbf{k}_j^{\mathrm{send},n}, \mathbf{e}_{ij}^n) \,, \\
\mathbf{k}_i^{\mathrm{rec},n+1} &= W_{\theta,\mathcal{V}}^n \, \mathbf{k}_i^{\mathrm{rec},n} + f_{\theta,\mathcal{V}}^n(\mathbf{k}_i^{\mathrm{rec},n}, \bigoplus_j \mathbf{e}_{ij}^{n+1}) \,.
\end{aligned} \tag{2}$$

During downsampling, receiver node embeddings $\mathbf{k}_i^{\mathrm{rec},n} \in \mathcal{V}^l$ are from level l, sender node embeddings $\mathbf{k}_j^{\mathrm{send},n} \in \mathcal{V}^{l+1}$ are from level $l{+}1$, and edge embeddings $\mathbf{e}_{ij}^n \in \mathcal{E}^{l,\mathrm{D}}$. During upsampling, the sender and receiver roles are reversed and $\mathbf{e}_{ij}^n \in \mathcal{E}^{l,\mathrm{U}}$. As aggregation $\bigoplus$ we use a max aggregation and $W_{\theta,\cdot}^n$ denotes weight matrices [8, 39, 40]. $f_{\theta,\cdot}^n$ denotes an MLPs that starts with layer normalization and consists of two linear layers with a hidden size of 128, with a Sigmoid Linear Unit (SiLU) activation [41] in between. Starting from the initial encoded embeddings ($n{=}0$), we apply a *V-cycle* that first updates the graph using a Pre-Processing stack and then compress information to the next coarser level via a Downsampling stack. We repeat this procedure until reaching the coarsest level $L = 3$. On the coarse graph, we apply a Pre-Processing stack followed by a Solving and a Post-Processing stack. During upsampling, we use an Upsampling stack and a Post-Processing stack at each level. We use 3 layers for Pre- and Post-Processing, 2 layers for Up- and Downsampling and 5 layers for the Solving stack. All stacks share their weights across levels, resulting in a total of 15 learnable layers. Given the final node embeddings $\mathbf{k}_i^N \in \mathcal{V}^0$ on the fine mesh, we apply a linear layer to decode the predicted displacement residual $\Delta\mathbf{u}_i^t$. We train ROBIN for a total of $500\,000$ iterations and with a batch size of 1 using ADAM [42] in *PyTorch* [43]. Training and testing are performed on an NVIDIA A100 GPU. We apply gradient clipping with an L_2-norm threshold of 1. The learning rate is linearly warmed up over the first $1\,000$ steps and then exponentially decayed from $1e{-}4$ to $1e{-}5$ over the remaining iterations. During training, we perturb the node positions $\mathbf{x}_i^t$ with Gaussian noise of standard deviation $10^{-5}\ \sigma_{\mathbf{x}}$ and the previous residual $\Delta\mathbf{u}_i^{t-1} = \mathbf{x}_i^t - \mathbf{x}_i^{t-1}$ with $10^{-6}\ \sigma_{\mathbf{x}}$ [9, 8], where $\sigma_{\mathbf{x}}$ denotes the standard deviation of the node positions.

Results

We compare AMPNs to FEM simulations using the Root Mean Squared Euclidean distance error $\text{RMSE} = \sqrt{1/(N_i N_j) \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} (\tilde{u}_{ij} - u_{ij})^2}$, where $\tilde{u}_{ij}$ and u_{ij} denote the predicted and ground-truth (FEM) node positions, respectively, with N_i nodes and N_j features. We then average the RMSE over time steps, the dataset, and finally report the mean μ and standard deviation σ across 3 random seeds.

Fig. 2 compares an AMPN-predicted stamp forming simulation to the FEM reference. AMPNs accurately reproduce the global deformation behavior, as shown in Fig. 2 a-b) after $70\,\%$ of the process time. Wrinkles at different scales match the wrinkles formation of the FEM simulation, and only minimal tool interactions are visible, consistent with the FEM. The FEM solution appears slightly noisier than the AMPN prediction, likely because the AMPNs focus on global solution accuracy and may miss very high-frequency components, as is common for autoregressive one step models [19, 21]. Fig. 2 c-d) shows the final process time step. Again, AMPNs closely match the FEM solution, correctly predicting ply draw-in and wrinkle formation. Tool intersections are comparable to the FEM, while very high spatial-frequency details differ slightly, with the AMPN prediction appearing less noisy overall.

We further compare AMPNs, which can be viewed as a special case of ROBIN with a single denoising step $K{=}1$, to ROBIN variants using more denoising steps ($K{=}5, 10, 20$). We use conventional inference rather than Rolling Diffusion-Batched Inference (ROBI) [21], since the dataset contains only 50 time steps. The potential speedup from ROBI is therefore less relevant, and we observe a significant accuracy improvement with conventional inference on this dataset.

Fig. 3 visualizes AMPNs and ROBIN predictions for different numbers of diffusion steps K and compares them to the FEM reference. The AMPNs most accurately captures the overall wrinkle formation. All models reproduce the material draw-in observed in the FEM simulation, while the $K = 5$ variant shows the largest deviation. In contrast, the $K = 10$ and $K = 20$ variants reproduce fine scale FEM behavior more realistically, for example the small wrinkles that appear near the center of the right boundary.

Table 1 summarizes the quantitative results for AMPNs and the different ROBIN variants. Consistent with Fig. 3, AMPNs achieve the lowest RMSE by most accurately capturing the global deformation. Moreover, a full simulation requires only 3.84 s on average on a GPU (NVIDIA A100). This enables real-time application, as the underlying forming process itself takes several seconds to complete. In contrast, a traditional FEM simulation requires 16.87 minutes on a CPU (Intel Xeon Gold 6230), so AMPNs achieve a speedup of more than two orders of magnitude. The runtime for ROBIN increases approximately linearly with the number of diffusion steps K, due to the growing number of model evaluations, while the RMSE of the $K{=}10$ and $K{=}20$ variants remains close to the RMSE of the AMPNs.

Table 1: Comparison of AMPNs and ROBIN ($K \in 5, 10, 20$). RMSE [mm] and runtime [s] reported as mean±std over the test set (3 seeds) on an NVIDIA A100 GPU. Speedup is relative to FEM (16.87 min on Intel Xeon Gold 6230).

Model	**AMPNs**	**ROBIN – 5**	**ROBIN – 10**	**ROBIN – 20**
RMSE [mm]	0.97 ± 0.07	1.18 ± 0.14	1.02 ± 0.05	1.03 ± 0.02
Runtime [s]	3.84 ± 0.02	18.38 ± 0.08	36.60 ± 0.11	73.07 ± 0.23
Speedup [-]	263.59	55.07	27.66	13.85

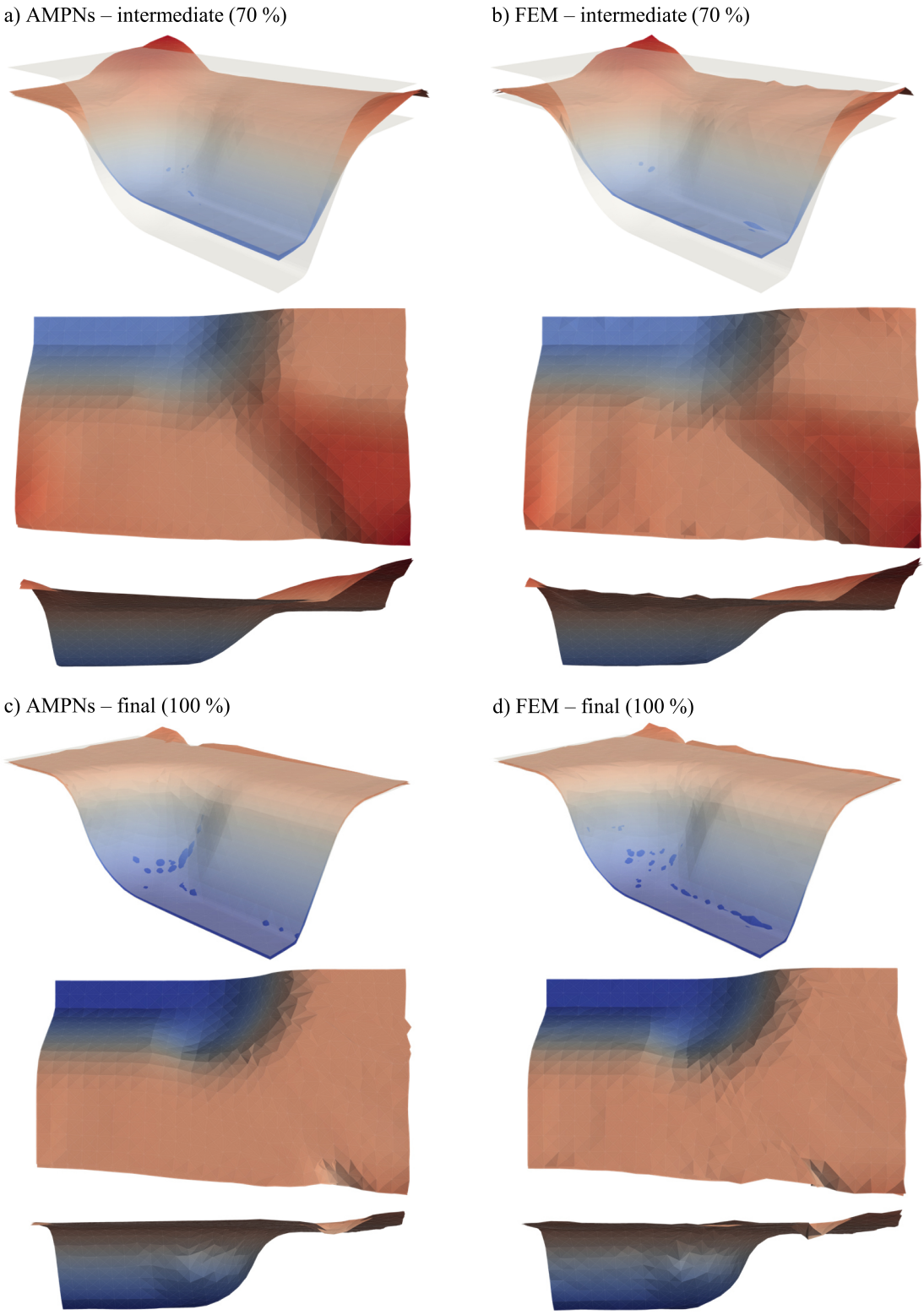

Fig. 2: Example unseen stamp forming simulation predicted by AMPNs compared to the FEM reference. a) and b) show the intermediate solution at 70 %, and c) and d) the final solution at 100 % (AMPNs vs. FEM). Colors indicate ply displacement in e_3-axis directions (red denotes large and blue small values), shown qualitatively to highlight wrinkles. Tools are displayed transparently. The tool–laminate intersection is visible from the high-contrast overlap of transparent tools and colored plies (plies in front appear strongly colored, plies behind are muted).

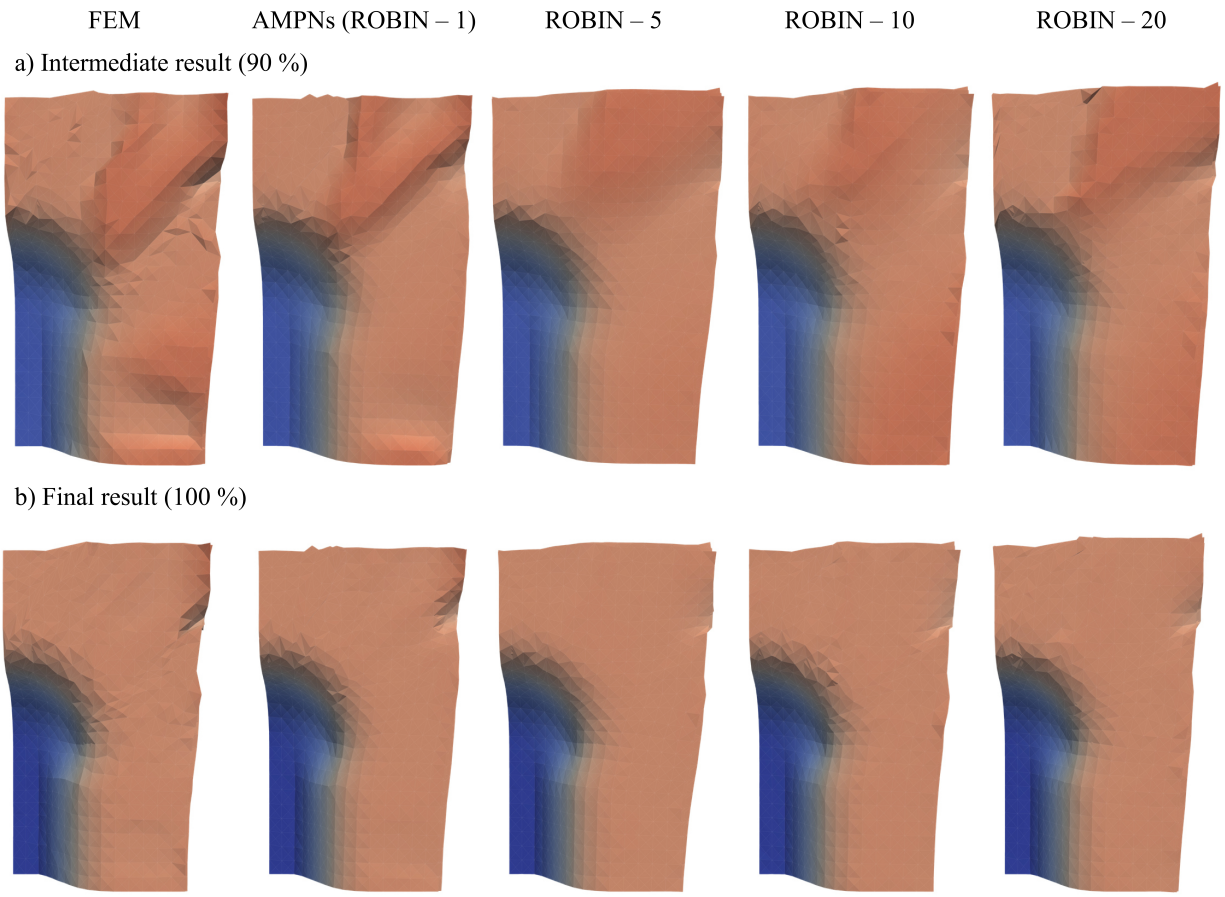

Fig. 3: AMPNs (i.e. ROBIN with $K = 1$ denosing steps) predictions and ROBIN predictions with different denoising steps $K = 5, 10, 20$ on unseen stamp forming simulations and to the FEM solution. a) shows the intermediate solution at 90 % and b) the final results (100 %). The color of each ply indicates its displacement in the $\boldsymbol{e}_3$-axis directions, shown qualitatively to highlight wrinkles. Red represents large values, and blue represents small values.

Conclusion and Outlook

In this work, we extended Algebraic-hierarchical Message Passing Networks (AMPNs) for stamp forming simulation of fiber-reinforced thermoplastic composite materials. AMPNs accurately reproduces local wrinkling formations and global material draw-in of state-of-the-art FEM simulations for unseen process settings and achieves a speedup of more than two orders of magnitude. We introduce component-wise and time-dependent contact edges for AMPNs to handle multi body contact dynamics in stamp forming simulations, which reduces the number of contact edges to the most important ones. We show that AMPNs, a Rolling Diffusion-Batched Inference Network (ROBIN) variant with a single diffusion step, outperform ROBIN variants with multiple diffusion steps in both accuracy and computational efficiency. Increasing the number of diffusion steps in ROBIN slightly improves fine-scale details but reduces runtime performance and overall prediction accuracy. This behavior contrasts with previous studies, in which more diffusion steps typically improved predictive accuracy. Analyzing the cause of this difference is an interesting avenue for future research. In this work, we considered only one geometry and one laminate setup, varying only the gripper configuration. However, learning general stamp forming models that can be applied to different geometries and material setups is promising because it allows us to evaluate new parts directly without retraining. Another interesting direction for future work is scaling ROBIN and more generally AMPNs to larger and more complex geometries.

Acknowledgements

This work is part of the DFG AI Research Unit 5339 (project no.459291153) and the DFG-funded Heisenberg project ”Digitalization of fiber-reinforced polymer processes for resource-efficient manufacturing of lightweight components” (project no. 455807141), which are both funded by the Deutsche Forschungsgemeinschaft, Germany (DFG, German Research Foundation). The authors acknowledge support by the state of Baden-Württemberg through bwHPC, as well as the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the German Federal Ministry of Education and Research. This work is supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition.

References

[1] Dibakar Bhattacharyya. *Composite sheet forming*, volume 11. Elsevier, 1997.

[2] Peng Wang, Nahiène Hamila, and Philippe Boisse. Thermoforming simulation of multilayer composites with continuous fibres and thermoplastic matrix. *Composites Part B: Engineering*, 52:127–136, 2013.

[3] Adil Wazeer, Apurba Das, Chamil Abeykoon, Arijit Sinha, and Amit Karmakar. Composites for electric vehicles and automotive sector: A review. *Green Energy and Intelligent Transportation*, 2(1), 2023.

[4] Mats G. Larson and Fredrik Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*, volume 10 of *Texts in Computational Science and Engineering*. Springer, Berlin, Heidelberg, 2013. ISBN 978-3-642-33286-9 978-3-642-33287-6.

[5] Philippe Boisse, Remko Akkerman, Pierpaolo Carlone, Luise Kärger, Stepan V. Lomov, and James A. Sherwood. Advances in composite forming through 25 years of ESAFORM. *International Journal of Material Forming*, 15(3):99, January 2022. ISSN 1960-6214. doi: 10.1007/s12289-022-01682-8.

[6] Philippe Boisse, Jin Huang, and Eduardo Guzman-Maldonado. Analysis and modeling of wrinkling in composite forming. *Journal of Composites Science*, 5(3):81, 2021.

[7] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional Neural Networks for Steady Flow Approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 481–490, San Francisco California USA, August 2016. ACM. ISBN 978-1-4503-4232-2.

[8] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*, October 2020.

[9] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to Simulate Complex Physics with Graph Networks. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8459–8468. PMLR, November 2020.

[10] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 00219991.

[11] Tobias Würth, Constantin Krauß, Clemens Zimmerling, and Luise Kärger. Physics-informed neural networks for data-free surrogate modelling and engineering optimization – An example from composite manufacturing. *Materials & Design*, 231:112034, July 2023. ISSN 02641275. doi: 10.1016/j.matdes.2023.112034.

[12] Han Gao, Luning Sun, and Jian-Xun Wang. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *Journal of Computational Physics*, 428:110079, March 2021. ISSN 00219991. doi: 10.1016/j.jcp.2020.110079.

[13] Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, April 2023. ISSN 0021-9991. doi: 10.1016/j.jcp.2023.111972.

[14] Han Gao, Matthew J. Zahr, and Jian-Xun Wang. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 390:114502, February 2022. ISSN 0045-7825. doi: 10.1016/j.cma.2021.114502.

[15] Tobias Würth, Niklas Freymuth, Clemens Zimmerling, Gerhard Neumann, and Luise Kärger. Physics-informed MeshGraphNets (PI-MGNs): Neural finite element solvers for non-stationary and nonlinear simulations on arbitrary meshes. *Computer Methods in Applied Mechanics and Engineering*, 429:117102, September 2024. ISSN 0045-7825.

[16] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, October 2020.

[17] Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message Passing Neural PDE Solvers. In *International Conference on Learning Representations*, October 2021.

[18] Jonas Linkerhägner, Niklas Freymuth, Paul Maria Scheikl, Franziska Mathis-Ullrich, and Gerhard Neumann. Grounding graph network simulators using physical sensor observations. In *International Conference on Learning Representations*, 2023.

[19] Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, December 2023.

[20] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling Diffusion Models. In *Forty-First International Conference on Machine Learning*, June 2024.

[21] Tobias Würth, Niklas Freymuth, Gerhard Neumann, and Luise Kärger. Diffusion-based hierarchical graph neural networks for simulating nonlinear solid mechanics. In *Advances in Neural Information Processing Systems*, 2025.

[22] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional Neural Processes. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1704–1713. PMLR, 2018. URL `https://proceedings.mlr.press/v80/garnelo18a.html`.

[23] Philipp Dahlinger, Tai Hoang, Denis Blessing, Niklas Freymuth, and Gerhard Neumann. Mango—adaptable graph network simulators via meta-learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[24] Philipp Dahlinger, Niklas Freymuth, Tai Hoang, Tobias Würth, Michael Volpp, Luise Kärger, and Gerhard Neumann. Context-aware learned mesh-based simulation via trajectory-level meta-learning. *arXiv preprint arXiv:2511.05234*, 2025.

[25] Steeven Janny, Aurélien Bénéteau, Madiha Nadri, Julie Digne, Nicolas Thome, and Christian Wolf. EAGLE: Large-scale Learning of Turbulent Fluid Dynamics with Mesh Transformers. In *The Eleventh International Conference on Learning Representations*, September 2022.

[26] Youn-Yeol Yu, Jeongwhan Choi, Woojin Cho, Kookjin Lee, Nayong Kim, Kiseok Chang, ChangSeung Woo, Ilho Kim, SeokWoo Lee, Joon Young Yang, Sooyoung Yoon, and Noseong Park. Learning Flexible Body Collision Dynamics with Hierarchical Contact Mesh Transformer. In *International Conference on Learning Representations*, October 2023.

[27] Kelsey Allen, Tatiana Lopez-Guevara, Kimberly L. Stachenfeld, Alvaro Sanchez Gonzalez, Peter Battaglia, Jessica B. Hamrick, and Tobias Pfaff. Inverse Design for Fluid-Structure Interactions using Graph Network Simulators. *Advances in Neural Information Processing Systems*, 35:13759–13774, December 2022.

[28] Clemens Zimmerling, Christian Poppe, Oliver Stein, and Luise Kärger. Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. *Materials & Design*, 214:110423, February 2022. ISSN 02641275.

[29] J. W. Ruge and K. Stüben. 4. Algebraic Multigrid. In Stephen F. McCormick, editor, *Multigrid Methods*, pages 73–130. Society for Industrial and Applied Mathematics, January 1987. ISBN 978-1-61197-188-0 978-1-61197-105-7.

[30] Petr Vanek, Jan Mandel, and Marian Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196, 1996.

[31] Dominik Dörr, Fabian J. Schirmaier, Frank Henning, and Luise Kärger. A viscoelastic approach for modeling bending behavior in finite element forming simulation of continuously fiber reinforced composites. *Composites Part A: Applied Science and Manufacturing*, 94:113–123, January 2017. ISSN 1359835X.

[32] Dominik Dörr, Markus Faisst, Tobias Joppich, Christian Poppe, Frank Henning, and Luise Kärger. Modelling Approach for Anisotropic Inter-Ply Slippage in Finite Element Forming Simulation of Thermoplastic UD-Tapes. In *AIP Proceedings of the 21th International ESAFORM Conference on Material Forming*, January 2018.

[33] Christian Poppe, Tobias Joppich, Dominik Dörr, Luise Kärger, and Frank Henning. Modeling and validation of gripper induced membrane forces in finite element forming simulation of continuously reinforced composites. In *AIP Conference Proceedings*, volume 1896, page 030002, October 2017.

[34] M. A. Khan, T. Mabrouki, E. Vidal-Sallé, and P. Boisse. Numerical and experimental analyses of woven composite reinforcement forming using a hypoelastic behaviour. Application to the double dome benchmark. *Journal of Materials Processing Technology*, 210(2):378–388, January 2010. ISSN 0924-0136.

[35] P. Harrison, R. Gomes, and N. Curado-Correia. Press forming a 0/90 cross-ply advanced thermoplastic composite using the double-dome benchmark geometry. *Composites Part A: Applied Science and Manufacturing*, 54:56–69, January 2013. ISSN 1359835X.

[36] Johannes Mitsch, Bastian Schäfer, and Luise Kärger. Rate-dependent 3D forming simulation of thermoplastic composite materials using visco-hyperelastic material modeling and 3D hexahedral solid-shell elements. *Composites Part A: Applied Science and Manufacturing*, 200:109306, January 2026. ISSN 1359-835X.

[37] Nathan Bell, Luke N. Olson, and Jacob Schroder. PyAMG: Algebraic multigrid solvers in python. *Journal of Open Source Software*, 7(LA-UR-23-26551), 2022.

[38] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[39] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, October 2018.

[40] Mario Lino Valencia, Tobias Pfaff, and Nils Thuerey. Learning Distributions of Complex Fluid Simulations with Diffusion Graph Networks. In *The Thirteenth International Conference on Learning Representations*, October 2024.

[41] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, November 2018. ISSN 08936080.

[42] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017.

[43] A. Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

The publication of this article was funded by the Karlsruhe Institute of Technology 10.13039/100009133