SPECFUSER: A HYBRID SPECTRAL TRANSFORMER FRAMEWORK FOR SOLID MECHANICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Solving solid mechanics problems of complex systems is notoriously demanding either in terms of time, computational resources, or data. Traditional numerical solvers require extensive modelling and fine-grained discretization for computational simulation. Scaling techniques serve as a fundamental strategy for managing computational overhead, including mass scaling which inflates material density, time scaling which compresses the event duration and so on. Nevertheless, a definitive trade-off between runtime and accuracy remains an inherent challenge. Deep learning solvers, on the other hand, are fast at inference, but they carry no physical prior and must recover the underlying mechanics from examples alone, which requires a lot of data and unreliable outside the distribution. To achieve the best of two worlds, we introduce SpecFuser, a hybrid spectral transformer framework that couples a numerical solver with a neural network. Specifically, the full-scaled model is first reduced by geometric scaling, and the numerical solver is run on this computationally cheaper, sub-scaled geometry. The resulting simulation is physically grounded yet geometrically compressed, capturing the essential low-frequency mechanical behaviour at a fraction of the original cost. The solver output, together with the geometric representations of both the sub-scaled and full-scale models, are then jointly encoded and passed to the network as structured input. A Fourier transform with a low-pass cut-off separates a physically exact low-frequency prior from scale-dependent residuals, the truncated spectrum is fused with spatial features and fed to a multi-head cross-attention transformer, which infers the full-scale solution. Because the solver has already supplied the low-frequency answer, the network is left with a low-dimensional residual rather than the full solution operator. This design for complex solid mechanics problems could not only significantly speed up the solving process, but also demonstrates superior capabilities in terms of generality and generalization.

1 INTRODUCTION

Solid mechanics problems lie at the core of many scientific and engineering disciplines (Lubarda & Lubarda, 2020), involving the analysis of how materials and structures respond to external loads, environmental changes, and internal defects (Barber & Klarbring, 2003). The complexity of these problems stems from the non-linear coupling between external force distributions and the resulting internal stress-strain fields, which are often dictated by complex boundary conditions and kinematic constraints (Holzapfel, 2002). In practical scenarios, predicting the progression of deformation and structural instability (Cedolin et al., 2010) requires an integrated analysis of how displacement gradients evolve across vast spatial and temporal scales (Kwon et al., 2008), and this multi-scale nature is particularly pronounced in multi-solid body large deformation problems (E, 2011). High-fidelity simulation of such systems, however, remains computationally prohibitive for industrial design cycles, where hundreds of configuration variants must be evaluated under tight time constraints.

Both existing classes of solvers struggle in these problems. Numerical solvers such as the finite element method (Zienkiewicz et al., 1977) are accurate, but capturing contact and springback requires fine meshes and implicit time integration, and a single full-scale run can take hours to days. Established acceleration techniques, including mass scaling (Olovsson et al., 2005), time scaling (Fish, 2013) and damping scaling (Chopra, 2007), act on the temporal or algorithmic side of the simulation and still trade accuracy for speed. Deep learning solvers, including neural operators (Li

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

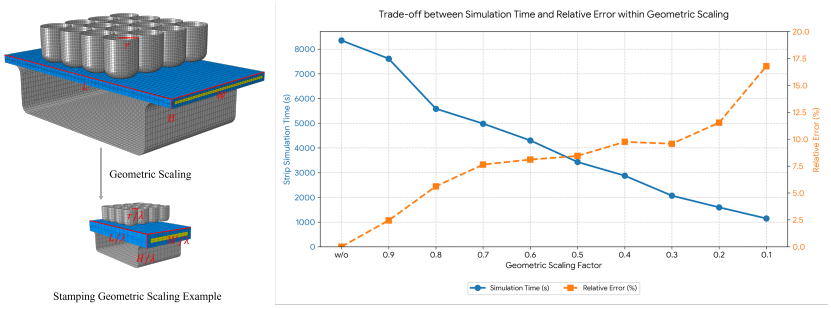

Figure 1: The Geometric Scaling Scenario. Illustration of geometric scaling in stamping simulations through a unified factor based on geometric representation as well as the trade-off between simulation time and relative error in stretch bending simulations.

et al., 2020a; Lu et al., 2021), graph networks (Pfaff et al., 2020) and Transformers (Wu et al., 2024), infer in milliseconds but carry no physical prior. They must learn the entire solution operator from examples, which requires large high-fidelity datasets and makes them unreliable outside the training distribution.

To address this gap, we propose **SpecFuser**, a hybrid framework in which combines the strength of the numerical solver and the neural network. Instead of simulating the full-scale system, we first reduce it by geometric scaling, every component is reduced by a unified factor in length, width and thickness, so that the proportions of the whole assembly are preserved as showed in Fig. 1. The numerical solver is run only on this inexpensive sub-scaled model. Its output, together with the geometric descriptors of both the sub-scaled and the full-scale modelling, is then passed to the network, which predicts the final full-scale configuration. Inside the network, a Fourier transform with a low-pass cut-off extracts the part of the sub-scaled solution that transfers across scales, a parallel spatial branch retains localized effects such as contact and local constraint. The Cross-attention processor afterwards uses the full-scale model descriptors as queries and sub-scaled features as keys and values, enabling the network to identify which sub-scaled modes are informative for the full-scale configuration. In this way, the neural network does not need to learn the entire non-linear mapping from scratch, it only needs to reconstruct the missing high-frequency information in the complement of the subspace where the numerical solver has already provided strong priors. At inference, the full-scale simulation is never run, the cost is one sub-scaled simulation plus one forward pass.

Geometric scaling targets the spatial dimension, a direct lever on cost that existing scaling techniques leave largely unused. As shown in Fig. 1, simulation time decreases monotonically with the scaling factor. Crucially, the sub-scaled run is still a complete physical simulation of the same process, with the same constitutive law, boundary conditions and contact, so its output is physically grounded in a way a purely learned surrogate is not. It cannot be used directly, however, simply rescaling the sub-scaled result by the scaling factor produces an error that grows as the factor decreases, because size effects break exact similarity. The role of the network is precisely to correct this discrepancy, not to solve the mechanics from scratch.

This hybrid architecture delivers multiple advantages. First, geometric scaling itself drastically reduces simulation time and resource consumption while establishing a reliable physical foundation for the full-scale model. Second, the Fourier transform frequency cut-off design filters out high-frequency noise and captures the fundamental deformation laws. Most importantly, because the learned correction operates on dimensionless modal coefficients rather than absolute-scale quantities, a single trained model can serve multiple scaling factors and generalize to unseen geometries with substantially less training data than purely data-driven methods require. This distinguishes SpecFuser from prior hybrid approaches, NCLaw (Ma et al., 2023) couples the network to the full-fidelity solver and therefore inherits its computational cost, while CFD-GCN (Belbute-Peres et al., 2020) embeds a solver but does not distinguish which parts of its output are trustworthy and which are noise, leaving the network to learn this distinction from data. Our method, by contrast, uses physical analysis from the outset to identify the transferable low-frequency prior, ensuring that the information handed to the network is inherently scale-covariant. Ultimately, the numerical solver guarantees accuracy and physical consistency for complex non-linear physics and deformation mech-

anisms, while deep learning provides computational efficiency for rapid non-linear mapping and real-time inference. High-fidelity simulation data obtained at reduced scales is accurately extrapolated to full-scale models through established scaling laws, enabling a seamless mapping from local mechanisms to macroscopic structural responses with orders-of-magnitude computational acceleration and robust multi-scale generalization capabilities.

2 Related Work

Deep learning is increasingly integrated with physical knowledge in hybrid solvers for solid mechanics, yet using an explicitly sub-scaled numerical simulation as the primary source of physical prior remains largely unexplored.

Numerical Solver Numerical methods such as the finite element method (FEM) (Zienkiewicz et al., 1977), finite volume method (FVM) (Belytschko et al., 2014), and finite difference method (FDM) (LeVeque, 2007) remain the main tools for simulating solid mechanics problems, but their accuracy and efficiency are constrained by computational cost, meshing complexity, convergence difficulties, and the treatment of nonlinear material behaviour (Rabczuk & Belytschko, 2007), particularly for large-scale, highly dynamic, or multi-physics problems (Fish & Shek, 2000). To balance efficiency and stability, scaling techniques are commonly employed, mass scaling (Olovsson et al., 2005) enlarges the stable time increment by artificially modifying density, time scaling (Barbero, 2023) compresses the physical event duration to reduce the number of increments, and damping scaling (Hughes, 2003) introduces controlled dissipation to suppress non-physical oscillations.

Deep Learning for Solid Mechanics Deep learning (DL) could address computational cost, constitutive modelling, and uncertainty quantification challenges in solid mechanics (Tao et al., 2025b). Operator learning maps between infinite-dimensional function spaces via neural networks (Lu et al., 2021; Li et al., 2020a; 2023c; Wu et al., 2024), while neural-FEM encodes initial conditions, governing equations, and boundary conditions as loss functions (Raissi et al., 2019; Maddu et al., 2022; Kharazmi et al., 2019; Liu et al., 2022). Neural operators based on CNNs (LeCun et al., 1995) such as FNO (Li et al., 2020a; Wen et al., 2022; Grady et al., 2022), GNNs (Scarselli et al., 2008) such as GNS (Sanchez-Gonzalez et al., 2020; Li et al., 2020b;c; Brandstetter et al., 2022), and Transformers (Vaswani et al., 2017; Wu et al., 2024; Li et al., 2023a; 2022) show promise for simple problems but may lack precision for complex systems and out-of-distribution scenarios. Neural-FEM approaches like PINNs (Raissi et al., 2019; Jagtap et al., 2020; Yu et al., 2022; Sheng & Yang, 2021) offer explainable physics guidance but require precise problem descriptions, constraining each design to specific instances with limited transferability.

Hybrid Solver The central challenge in hybrid approaches is how to integrate physical knowledge with data-driven networks. NCLaw (Ma et al., 2023) combines neural networks with numerical solvers through iterative cross-updating, and while this generalizes across unseen initial conditions, boundary conditions, and geometries, it inherits the cost of the full-fidelity solver, the network operates at the same scale as the target problem, so the full solution must be computed at every iteration. CFD-GCN (Belbute-Peres et al., 2020) embeds a complete numerical solver and processes its output with a GCN (Kipf & Welling, 2016), but treats that output as a black box, leaving the network to distinguish signal from noise entirely from data without any analytical understanding of the error structure, moreover, its GCN design and Euler formulation, tailored to fluid dynamics, are difficult to transfer to Lagrangian solid mechanics, where material points are tracked rather than fixed spatial locations.

SpecFuser belongs to this family but departs in two respects. First, the solver runs only on a geometrically sub-scaled model, avoiding rather than inheriting the full-scale cost. Second, rather than consuming solver output uniformly, it separates it in the frequency domain into a scale-transferable low-frequency prior and a scale-dependent residual, and operates on a Lagrangian representation suited to large deformation and multi-body contact.

3 Methodology

SpecFuser predicts full-scale outcomes from a computationally cheap sub-scaled simulation, with a neural network correcting the scale-dependent residual in a spectrally filtered space. We formalize

the task in §3.1, analyse geometric scaling in the Fourier domain to motivate the low-pass cut-off in §3.2, present the Encoder and Processor in §3.3, and describe two case studies in §3.4.

3.1 Problem Description

We formulate the problem of predicting full-scale simulation outcomes from sub-scale models for multi-body manufacturing systems involving multiple deformable solids, rigid solids, and loads. Given a multi-body system $\mathcal{S}$ comprised of different solid objects, we adopt a Lagrangian perspective where the system is discretized into a finite set of mesh points representing these entities.

Inputs: The inputs consist of (1) geometric representations of all bodies in the system at both full-scale and sub-scale, and (2) loading conditions and boundary constraints. Specifically, the system state is characterized by the collective configuration of N^d deformable bodies, denoted as $\mathbf{u}^d = \{\mathbf{u}_i^d \in \mathbb{R}^{N_i^d \times 3}\}_{i=1}^{N^d}$, N^r rigid bodies represented by $\mathbf{u}^r = \{\mathbf{u}_i^r \in \mathbb{R}^{N_i^r \times 3}\}_{i=1}^{N^r}$, and N^l loads defined as $\mathbf{u}^l = \{\mathbf{u}_i^l \in \mathbb{R}^{N_i^l \times 3}\}_{i=1}^{N^l}$ when yielded by prescribed movements of solids, or otherwise specified case by case.

Outputs: The output is the predicted final configuration of deformable bodies $\hat{\mathbf{u}}_{full}^d$ at full-scale after the complete forming process, including both the loading phase and the springback phase.

Objective: Given the simulation results from a computationally inexpensive sub-scale model and the geometric descriptors of the full-scale system, our goal is to accurately predict the full-scale deformation outcomes without running the expensive full-scale simulation. This enables rapid design iteration and optimization in industrial manufacturing processes.

3.2 Geometric Scaling Analysis

We leverage geometric scaling to transform the full-scaled model into the sub-scaled model in our model design. Through geometric scaling analysis, it brings the foundation of capturing the structural deformation laws from the sub-scaled model for the prediction of the full-scaled model, leveraging this we design the two branch inputs as well as the frequency cut off of our model. To bridge the two representations, Fourier spectral decomposition (Bracewell & Bracewell, 1986; Stein & Shakarchi, 2011; Oppenheim, 1999) is utilized to capture the geometric similarity between the two models.

Giving the spatial geometry of the original component of length L as a signal $f(x)$, where $x \in [0, L]$. Consider a scaling factor λ such that the geometry of the sub-scaled model is represented by $f_\lambda(x) = f(x/\lambda)$ for $x \in [0, L/\lambda]$. According to the Similarity Theorem (or Scaling Property) (Papoulis & Maradudin, 1963) of the Fourier Transform, the relationship between their respective spectral representations $F(\omega)$ and $F_\lambda(\omega)$ is given by

$$\mathcal{F}\{f(x/\lambda)\} = F_\lambda(\omega) = \lambda F(\lambda\omega)$$

This identity reveals two critical scaling behaviors: first, a frequency shift, where the characteristic spectral peaks of the geometry move toward higher frequencies by a factor of λ; and second, magnitude scaling, where the spectral density is scaled by λ to maintain energy conservation in the transform domain.

To decouple the geometric features from the absolute physical scale, we introduce the dimensionless coordinate $\xi = x/L$ (where $\xi \in [0, 1]$). The geometry is then expressed via a Fourier series

$$f(\xi) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n\xi}$$

In an ideal similarity paradigm, the complex coefficients C_n which representing the n-th harmonic mode should remain invariant across scales, as they characterize the intrinsic shape and relative deformation patterns.

In the context of multi-body simulations and experimental measurements, frequency-domain decomposition serves as the foundational inspiration for our model architecture. We characterize the captured signal $f(x)$ as a superposition of multiscale physical phenomena, decomposed into low- and high-frequency components:

$$f(x) = f_{low}(x) + f_{high}(x)$$

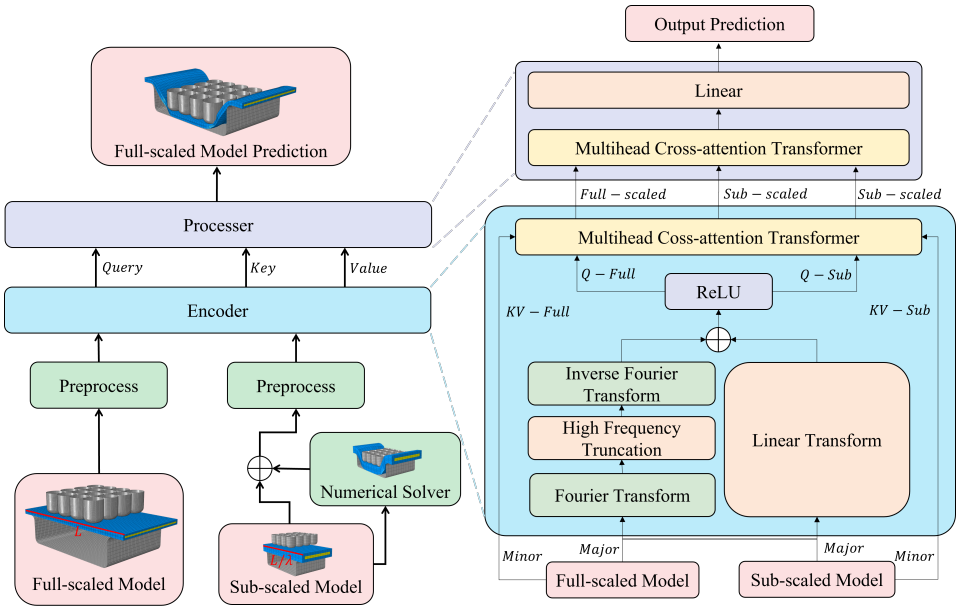

Figure 2: Overall architecture of SpecFuser. The full architecture is composed by a numerical solver followed by Encoder and Processor.

Here, the low-frequency component f_{low} is governed by macro-mechanical boundary conditions and constitutive laws, representing global curvature and elastic springback. These large-scale deformation patterns exhibit strong self-similarity across different scaling factors λ due to the inherent scale-invariance of elastic continuum mechanics (Katsuragi et al., 2016; Buckingham, 1914; Sedov et al., 1960). Conversely, the high-frequency component f_{high} is associated with non-scalable physical factors, such as material grain size (Argibay et al., 2017), localized friction fluctuations (Chan & Fu, 2012; Wang et al., 2014), and micro-vibrations within the tooling—phenomena that become increasingly dominant as geometric scale decreases (Vollertsen et al., 2004; Engel & Eckstein, 2002).

By applying a low-pass filter to truncate the spectrum beyond a critical normalized frequency n_c, the filtered signal $\hat{f}(\xi)$ focuses exclusively on structural modes. This truncation effectively suppresses scale-independent noise, thereby significantly enhancing the correlation between the sub-scaled and the full-scale model.

3.3 Model Overview

As shown in Fig 2, SpecFuser couples a numerical solver which is Abaqus in this work (Abaqus, 2011; Barbero, 2023) with an Encoder and a Processor, the solver is treated as a swappable component and can be replaced by any instance suited to future implementations. During training, the solver simulates the system at both scales. Guided by geometric priors, we then extract major features, such as macroscopic deformation characteristics, and minor features, such as cross-sectional geometries and system parameters. The Encoder transforms and fuses these features with spatial and spectral information, and the Processor learns the mapping from the sub-scaled to the full-scaled paradigm. At inference, no full-scale simulation is run, given only the geometric descriptors and the sub-scaled results, SpecFuser predicts the full-scale outcome directly.

3.3.1 Encoder

The Encoder block designed is to extract and fuse the spatial temporal features of the multi-solid-body system. Let $\mathbf{H}_{full} \in \mathbb{R}^{C\times L}$ and $\mathbf{H}_{sub} \in \mathbb{R}^{C\times L}$ denote the primary feature representations from the full-scaled and sub-scaled models respectively. Both the major features of the full-scaled and the sub-scaled model are first fed into two parallel processing paths inspired by the Fourier Neural Operator architecture to capture multi-scale characteristics within a unified transformation

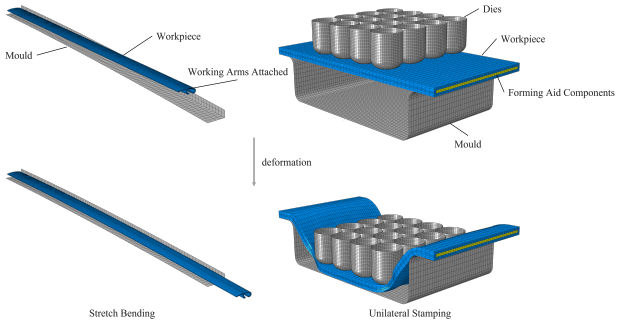

Figure 3: The case study scenarios including stretch bending and unilateral stamping.

space. Specifically, for an input feature $\mathbf{H} \in \{\mathbf{H}_{full}, \mathbf{H}_{sub}\}$, the transformation is formulated as:

$$\mathbf{Z}_{major} = \sigma\left(\mathbf{W}_s\mathbf{H} + \mathcal{F}^{-1}\left(R \cdot \mathcal{F}(\mathbf{H})\right)\right)$$

where the spatial branch applies a linear transformation $\mathbf{W}_s \in \mathbb{R}^{C\times C}$ to preserve high-resolution spatial structures and local interactions, while the spectral branch performs a sequence of Fourier Transform ($\mathcal{F}$), a learnable spectral weight multiplication (R), and an Inverse Fourier Transform ($\mathcal{F}^{-1}$). This spectral path is dedicated to capturing global dependencies and long-range correlations by modeling the system's dynamics in the frequency domain. The outputs from these two branches are integrated through an element-wise summation and passed through a ReLU activation layer (σ) to introduce non-linearity and refine the integrated representation $\mathbf{Z}_{major}$. Following the extraction, the primary features from both scales are integrated with the minor features $\mathbf{Z}_{minor}$ through a multi-head cross-attention transformer. In this mechanism, the queries $\mathbf{Q}$ are projected from the fused representation $\mathbf{Z}_{major}$, while the keys $\mathbf{K}$ and values $\mathbf{V}$ are derived from the minor features $\mathbf{Z}_{minor}$. This ensures that the model adaptively weights the most relevant features across the entire multiscale spectrum, ensuring high-fidelity feature alignment. Finally, the Encoder outputs the refined

$$\mathbf{F}_{full} = \text{MHCA}(\mathbf{Z}^{full}_{major}, \mathbf{Z}^{full}_{minor}, \mathbf{Z}^{full}_{minor})$$

and

$$\mathbf{F}_{sub} = \text{MHCA}(\mathbf{Z}^{sub}_{major}, \mathbf{Z}^{sub}_{minor}, \mathbf{Z}^{sub}_{minor})$$

to the downstream Processer, facilitating the final prediction for the multi-solid-body system.

3.3.2 PROCESSOR

The Processor block serves as the terminal decoding stage of the architecture, responsible for translating the fused multiscale representations into final physical predictions. The Processor receives the refined features from the Encoder and further aligns them through a Multihead Cross-attention Transformer. This layer is specifically configured to integrate the descriptors from different modelling—taking the full-scaled modelling features as the Query and the sub-scaled modelling features as the Key and Value—to ensure that the full-scaled dynamics are informed by high-fidelity sub-scaled details. The cross-attention operation within the Processor is defined as:

$$\mathbf{F} = \text{MHCA}(\mathbf{F}_{full}, \mathbf{F}_{sub}, \mathbf{F}_{sub})$$

where $\mathbf{F}_{full}$ and $\mathbf{F}_{sub}$ represent the different modelling features produced by the preceding stage. Following this hierarchical alignment, the consolidated features $\mathbf{F}$ are passed through a Linear layer, which acts as a prediction head to map the latent embeddings back to the physical state space. The final output prediction $\hat{\mathbf{Y}}$ is formulated as:

$$\hat{\mathbf{Y}} = \mathbf{W}_{out}\mathbf{Z} + \mathbf{b}_{out}$$

where $\mathbf{W}_{out}$ and $\mathbf{b}_{out}$ are learnable parameters. By leveraging this sequential attention and projection strategy, the Processor effectively decodes the complex spatiotemporal interactions of the multi-solid-body system, yielding high-fidelity predictions that maintain consistency across multiple modelling.

3.4 CASE STUDY

We validate SpecFuser on two representative metal forming scenarios, stretch bending and unilateral stamping. These cases are deliberately chosen because they represent computationally demanding industrial problems where high-fidelity simulation is essential but prohibitively expensive. Both processes involve complex multi-body interactions, large elastic-plastic deformations, and springback behaviour that requires fine mesh resolution and implicit time integration to capture accurately. A single full-scale simulation can take hours to days of computational time, making iterative design optimization impractical. This makes them ideal benchmarks for our approach, successful prediction of full-scale outcomes from sub-scale simulations in these challenging scenarios would validate the potential for substantial reductions in computational cost while maintaining accuracy in industrial manufacturing applications.

Stretch Bending. Large deformation problems of elastic-plastic solids are among the most representative challenges in solid mechanics (Bathe & Ozdemir, 1976). We choose stretch bending (Clausen et al., 2000) as one of our case studies. As shown in Fig 3, the process predicts the final shape of a metal workpiece by analyzing its deformation over a mould during loading and its subsequent springback after unloading. For this problem, we create the Strip dataset containing 1000 sample pairs. Each sample pair includes both a full-scaled and a sub-scaled modelling, each comprising the characteristic line and cross-section of the 3D workpiece, the characteristic line of the 3D mould, the motion parameters, and the springback characteristic line. The modellings are generated following procedures similar to the LaDEEP dataset (Tao et al., 2025a), including 5 different representative types of cross-sections with randomly sampled workpiece and mould geometries. The motion parameters are calculated accordingly, and the springback results are computed using Abaqus (Abaqus, 2011; Barbero, 2023). The sub-scaled modellings are derived from the full-scaled modellings through geometric scaling of the system dimensions. To fit the specific geometric representation, we define the workpiece by a 2D cross-section image of $1 \times H \times W$ and a characteristic line of $N \times 3$. The motion parameters are specified by $\{u_x, u_y, u_z, r_x, r_y, r_z\}$, where $\{u_x, u_y, u_z\}$ represent three translational degrees of freedom and $\{r_x, r_y, r_z\}$ represent three rotational degrees of freedom.

Unilateral Stamping. Another important scenario of multi-solid systems is unilateral stamping, which we choose as our second case study (Li et al., 2002). As shown in Fig 3, the process simulates the deformation of a hollow metal workpiece using multiple rigid stamping dies to achieve the target shapes. For this problem, we create the Stamping dataset containing 50000 sample pairs of full-scaled and sub-scaled modellings. Each modelling includes the characteristic line and cross-section of the 3D workpiece, the forming aid component, the mould, and the stamping dies. The modellings are generated following procedures similar to the Unisoma Unilateral Stamping dataset (Tao et al., 2025b), which simulates the deformation of a hollow metal workpiece via a multi-die system. To ensure cross-sectional integrity during high-pressure stamping, a rubber insert is integrated into the cavity as a supportive internal medium. The final springback results are also calculated by Abaqus (Abaqus, 2011; Barbero, 2023). The sub-scaled modellings are derived from the full-scaled modellings through geometric scaling of the system dimensions. Within this specific geometric representation, we define the workpiece by a cross-section of $H^w \times W^w$ mesh points $N_s^w \times 3$ and a characteristic line of $N_l^w \times 3$. The forming aid component is defined similarly by a cross-section of $H^r \times W^r$ mesh points $N_s^r \times 3$ and a characteristic line of $N_l^r \times 3$. The loads of the unilateral stamping system are defined by a set of stamping dies, denoted by a cross-section of $H^d \times W^d$ mesh points $N_s^d \times 3$ and a characteristic line of $N_l^d \times 3$.

4 EXPERIMENTS

We evaluate the SpecFuser compared with multiple graph-based, operator-based and transformer-based methods on the Strip and the Stamping datasets, as well as the efficiency analysis.

4.1 EXPERIMENTAL SETTINGS

Baselines The baselines are mainly from three framework groups which are graph-based, operator-based and transformer-based architectures. Representative, graph-based models including MeshGraphNet(MGN) (Pfaff et al., 2020), GraphSAGE (Hamilton et al., 2017), Hierarchical Contact

Table 1: Performance comparison on Strip and Stamping datasets. Relative L_2 error is reported (smaller is better), as mean $\pm$ std over 5 random seeds. The best results are highlighted in bold.

Model	Strip	Stamping
MGN	0.0123 ± 0.0002	0.0248 ± 0.0003
GraphSAGE	0.0121 ± 0.0005	0.0305 ± 0.0006
HCMT	0.0196 ± 0.0006	0.0258 ± 0.0003
FNO	0.0062 ± 0.0003	0.0238 ± 0.0006
Geo-FNO	0.0055 ± 0.0004	0.0566 ± 0.0003
LSM	0.0195 ± 0.0003	0.0384 ± 0.0004
Transolver	0.0083 ± 0.0003	0.0206 ± 0.0002
FactFormer	0.0051 ± 0.0002	0.0185 ± 0.0003
SpecFuser	$\mathbf{0.0028 \pm 0.0002}$	$\mathbf{0.0137 \pm 0.0004}$

Mesh Transformer (HCMT) (Yu et al., 2023), operator-based models including Fourier Neural Operator (FNO) (Li et al., 2020a), Geometry-Aware Fourier Neural Operator (Geo-FNO) (Li et al., 2023b), Latent Spectral Models (LSM) (Wu et al., 2023) and transformer-based models including Tansolver (Wu et al., 2024), FactFormer (Li et al., 2023a).

Implementation For SpecFuser, we set modes, width, embedding dimension, number of heads, feed forward dimension and number of layers to 16, 64, 64, 8, 1024 and 4 respectively across all experiments. All experiments are conducted on a single RTX 3090 GPU. We utilize Relative L_2 as the evaluation metrics for the prediction.

4.2 Results

Accuracy Performance To evaluate the performance of SpecFuser, we conducted a comparative analysis against several state-of-the-art models, including graph-based methods (MGN (Pfaff et al., 2020), GraphSAGE (Hamilton et al., 2017), HCMT (Yu et al., 2023)), operator learning solvers (FNO (Li et al., 2020a), Geo-FNO (Li et al., 2023b), LSM (Wu et al., 2023)), and transformer-based architectures (Transolver (Wu et al., 2024), FactFormer (Li et al., 2023a)). The experiments were performed on the two generated datasets. Table 1 summarizes the quantitative results in terms of relative L_2 error. SpecFuser achieves the lowest relative L_2 error across both datasets, outperforming all baseline models. This demonstrates its robust capability in capturing complex physical dynamics and geometric features. While traditional models like FNO (Li et al., 2020a) and FactFormer (Li et al., 2023a) provide competitive baselines, SpecFuser consistently maintains a performance edge. Its architectural integration of spectral features and transformer-based global attention allows for a more precise reconstruction of the solution fields. The consistent success on both the Strip and the Stamping datasets suggests that the proposed hybrid spectral-transformer approach is versatile and can effectively handle different boundary conditions and physical configurations. The experimental results suggests that SpecFuser offers noticeable improvements in predictive accuracy over several established methods in these specific scenarios. These findings highlight the effectiveness of integrating spectral features with transformer architectures for this class of problems.

Time Efficiency To evaluate the computational advantages of the proposed hybrid architecture, we compared the execution times for the Strip and Stamping datasets across various geometric scaling factors. The baseline is defined by a full-scaled pure Abaqus simulation while our hybrid approach accounts for the cumulative time of the sub-scaled Abaqus simulation and the neural network inference. As shown in Table 2, the geometric scaling strategy effectively mitigates the computational bottleneck of the numerical solver. Decreasing the scaling factor from 0.9 to 0.1 yields a substantial reduction in total processing time, confirming that geometric simplification significantly accelerates complex simulations. By leveraging the SpecFuser to bypass high-resolution numerical solver's requirements, the framework maintains high fidelity while consuming only a fraction of the baseline temporal resources. Furthermore, since the average inferencing time for SpecFuser on Strip

Table 2: Time comparison on Strip and Stamping datasets. Results in seconds are reported in sum of Abaqus simulation time for geometric scaling applied and SpecFuser inferencing time, or pure Abaqus time for the baseline, the average inferencing time for SpecFuser on Strip and Stamping are 0.017s and 0.015s.

Scaling Factor	Strip (s)	Stamping (s)	Scaling Factor	Strip (s)	Stamping (s)
w/o	8347.44	789.71	0.5	3430.54	374.48
0.9	7611.79	741.97	0.4	2875.32	309.00
0.8	5585.13	564.25	0.3	2066.14	289.09
0.7	4981.80	482.50	0.2	1594.56	277.79
0.6	4301.70	439.62	0.1	1144.48	230.85

and Stamping are 0.017s and 0.015s, the millisecond-level inference overhead is negligible compared to the time saved during the simulation phase. This net gain in end-to-end efficiency enables rapid design iterations and supports real-time optimization in industrial metal-forming applications, providing a robust mechanism for time compression without compromising physical validity.

Ablation Study To analyse the sensitivity of SpecFuser to the spectral cut-off hyperparameter, we vary modes n_c while keeping all other settings at their defaults (width = 64, 3 seeds each).

Table 3: Ablation on spectral cut-off modes n_c. Relative L_2 error (mean $\pm$ std, 3 seeds). Default setting is in **bold**.

Modes n_c	Strip	Stamping
4	0.0132 ± 0.0005	0.0285 ± 0.0006
8	0.0063 ± 0.0003	0.0179 ± 0.0006
16	$\mathbf{0.0028 \pm 0.0002}$	$\mathbf{0.0137 \pm 0.0004}$
32	0.0041 ± 0.0003	0.0151 ± 0.0005

The results reveal a clear optimum at $n_c = 16$. Reducing n_c below this threshold progressively discards low-frequency deformation modes essential for predicting global curvature and springback, raising the error substantially. Increasing n_c beyond 16 reintroduces scale-dependent high-frequency components into the spectral prior, adding noise that the network must correct rather than exploit and causing a mild but consistent performance drop on both datasets. This validates the theoretical analysis in Section 3.2 which the low-pass cut-off must span the scale-transferable band without admitting scale-corrupted frequencies.

5 Conclusion

In this paper, we present SpecFuser, a hybrid spectral transformer framework that couples a numerical solver with a neural network to address the computational cost and generalization challenges of complex solid mechanics problems. The framework combines geometric scaling with a dual-module Encoder–Processor architecture, the Encoder fuses spatial transformations with spectral global dependencies to capture multi-scale spatiotemporal features, and the Processor uses cross-attention to map sub-scaled features to full-scale predictions. Through Fourier spectral decomposition, we showed that sub-scaled simulations provide robust physical priors that substantially reduce the burden on the numerical solver while preserving the low-frequency structural modes that transfer across scales.

Experiments on the Strip (stretch bending) and Stamping (unilateral stamping) datasets confirm that SpecFuser consistently outperforms leading graph-based, operator-based, and transformer-based models in predictive accuracy, while reducing end-to-end cost by orders of magnitude through reduced-scale simulation without sacrificing precision. Future work will extend the architecture to more diverse multi-physics coupling problems and more complex out-of-distribution scenarios.

AI USE STATEMENT

Generative AI tools were used in a supporting capacity during this work. They assisted in polishing and revising the manuscript text, in retrieving relevant literature, and in assisting auxiliary code for experimental workflows. The core ideas, problem formulation, and algorithmic design were developed by the authors. All AI-assisted text was reviewed, fact-checked, and revised by the authors, and every cited reference was independently located and verified against the original publication. The authors take full responsibility for the final content of this work, including all AI-assisted text, citations, and code.

REFERENCES

G Abaqus. Abaqus 6.11. *Dassault Systemes Simulia Corporation, Providence, RI, USA*, 3:73, 2011.

Nicolas Argibay, M Chandross, Shengfeng Cheng, and Joseph R Michael. Linking microstructural evolution and macro-scale friction behavior in metals. *Journal of materials science*, 52(5):2780–2799, 2017.

JR Barber and Anders Klarbring. Solid mechanics and its applications. In *Solid mechanics and its applications*. Springer, 2003.

Ever J Barbero. *Finite element analysis of composite materials using Abaqus®*. CRC press, 2023.

Klaus-Jürgen Bathe and Haluk Ozdemir. Elastic-plastic large deformation static and dynamic analysis. *Computers & Structures*, 6(2):81–92, 1976.

Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In *international conference on machine learning*, pp. 2402–2411. PMLR, 2020.

Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2014.

Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-hill New York, 1986.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Edgar Buckingham. On physically similar systems; illustrations of the use of dimensional equations. *Physical review*, 4(4):345, 1914.

Luigi Cedolin et al. *Stability of structures: elastic, inelastic, fracture and damage theories*. World Scientific, 2010.

Wai Lok Chan and MW Fu. Experimental and simulation based study on micro-scaled sheet metal deformation behavior in microembossing process. *Materials Science and Engineering: A*, 556: 60–67, 2012.

Anil K Chopra. *Dynamics of structures*. Pearson Education India, 2007.

Arild H Clausen, Odd S Hopperstad, and Magnus Langseth. Stretch bending of aluminium extrusions for car bumpers. *Journal of Materials Processing Technology*, 102(1-3):241–248, 2000.

Weinan E. *Principles of multiscale modeling*. Cambridge University Press, 2011.

Ulf Engel and Ralf Eckstein. Microforming—from basic research to its realization. *Journal of Materials Processing Technology*, 125:35–44, 2002.

Jacob Fish. *Practical multiscaling*. John Wiley & Sons, 2013.

Jacob Fish and Kamlun Shek. Multiscale analysis of composite materials and structures. *Composites Science and Technology*, 60(12-13):2547–2556, 2000.

TJ Grady, Rishi Khan, Mathias Louboutin, Ziyi Yin, Philipp A Witte, Ranveer Chandra, Russell J Hewett, and Felix J Herrmann. Towards large-scale learned solvers for parametric pdes with model-parallel fourier neural operators. *arXiv preprint arXiv:2204.01205*, 2022.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Gerhard A Holzapfel. Nonlinear solid mechanics: a continuum approach for engineering science, 2002.

Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2003.

Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

Hiroaki Katsuragi et al. *Physics of soft impact and cratering*, volume 910. Springer, 2016.

Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*, 2019.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Young W Kwon, David H Allen, and Ramesh Talreja. *Multiscale modeling and simulation of composite materials and structures*, volume 47. Springer, 2008.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

Ming-Zhe Li, Zhong-Yi Cai, Zhen Sui, and QG Yan. Multi-point forming technology for sheet metal. *Journal of Materials Processing Technology*, 129(1-3):333–338, 2002.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for pde surrogate modeling. *Advances in Neural Information Processing Systems*, 36:28010–28039, 2023a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020c.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023b.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023c.

Songming Liu, Hao Zhongkai, Chengyang Ying, Hang Su, Jun Zhu, and Ze Cheng. A unified hard-constraint framework for solving geometrically complex pdes. *Advances in Neural Information Processing Systems*, 35:20287–20299, 2022.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Marko V Lubarda and Vlado A Lubarda. *Intermediate solid mechanics*. Cambridge University Press, 2020.

Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B Tenenbaum, Tao Du, Chuang Gan, and Wojciech Matusik. Learning neural constitutive laws from motion observations for generalizable pde dynamics. In *International Conference on Machine Learning*, pp. 23279–23300. PMLR, 2023.

Suryanarayana Maddu, Dominik Sturm, Christian L Müller, and Ivo F Sbalzarini. Inverse dirichlet weighting enables reliable training of physics informed neural networks. *Machine Learning: Science and Technology*, 3(1):015026, 2022.

Lars Olovsson, Kjell Simonsson, and Mattias Unosson. Selective mass scaling for explicit finite element analyses. *International Journal for Numerical Methods in Engineering*, 63(10):1436–1445, 2005.

Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

Athanasios Papoulis and AA Maradudin. The fourier integral and its applications, 1963.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Timon Rabczuk and T23253911128 Belytschko. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer methods in applied mechanics and engineering*, 196(29-30):2777–2799, 2007.

Maziar Raissi, Paris Perdikaris, and George 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, 2019.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Leonid Ivanovich Sedov, Morris Friedman, and RE Street. Similarity and dimensional methods in mechanics. *Physics Today*, 13(9):50–52, 1960.

Hailong Sheng and Chao Yang. Pfnn: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. *Journal of Computational Physics*, 428:110085, 2021.

Elias M Stein and Rami Shakarchi. *Fourier analysis: an introduction*, volume 1. Princeton University Press, 2011.

Shilong Tao, Zhe Feng, Haonan Sun, Zhanxing Zhu, and Yunhuai Liu. Ladeep: A deep learning-based surrogate model for large deformation of elastic-plastic solids. *arXiv preprint arXiv:2506.06001*, 2025a.

Shilong Tao, Zhe Feng, Haonan Sun, Zhanxing Zhu, and Yunhuai Liu. Unisoma: A unified transformer-based solver for multi-solid systems. *arXiv preprint arXiv:2506.06021*, 2025b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Frank Vollertsen, Zhenyu Hu, H Schulze Niehoff, and Carmen Theiler. State of the art in micro forming and investigations into micro deep drawing. *Journal of Materials Processing Technology*, 151(1-3):70–79, 2004.

Chunju Wang, Bin Guo, and Debin Shan. Friction related size-effect in microforming–a review. *Manufacturing Review*, 1:23, 2014.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

Haixu Wu, Tengge Hu, Huakun Luo, Jianmin Wang, and Mingsheng Long. Solving high-dimensional pdes with latent spectral models. *arXiv preprint arXiv:2301.12664*, 2023.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.

Youn-Yeol Yu, Jeongwhan Choi, Woojin Cho, Kookjin Lee, Nayong Kim, Kiseok Chang, Chang-Seung Woo, Ilho Kim, Seok-Woo Lee, Joon-Young Yang, et al. Learning flexible body collision dynamics with hierarchical contact mesh transformer. *arXiv preprint arXiv:2312.12467*, 2023.

Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Perumal Nithiarasu, and JZ Zhu. *The finite element method*, volume 3. Elsevier, 1977.