Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 416 (2023) 116343

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

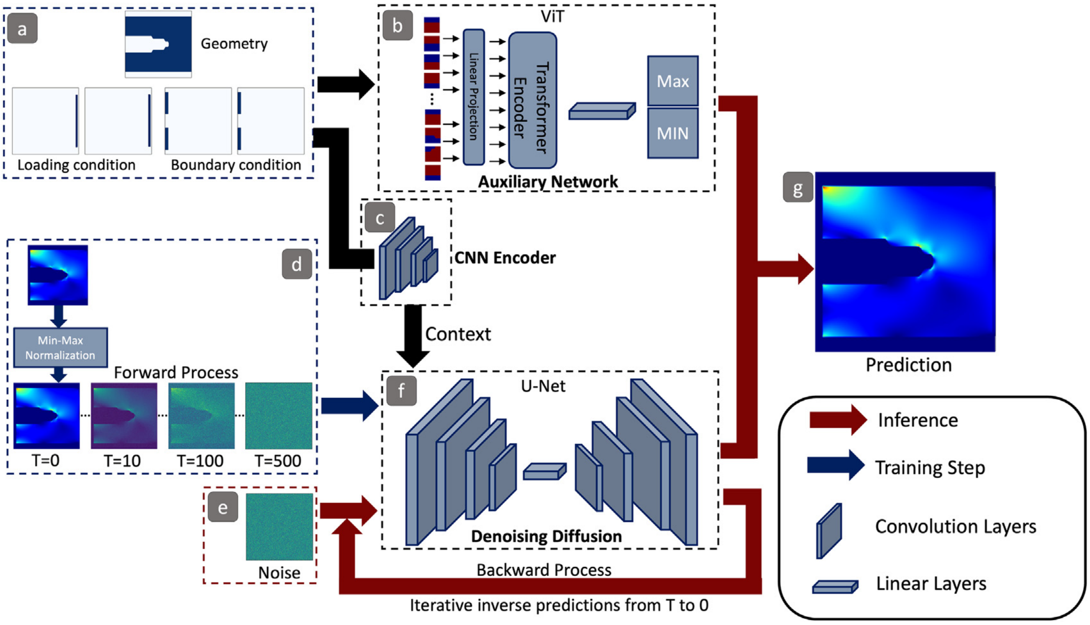
StressD: 2D Stress estimation using denoising diffusion model

Yayati Jadhav, Joseph Berthel, Chunshan Hu, Rahul Panat, Jack Beuth, Amir Barati Farimani*

Mechanical Engineering, Carnegie Mellon University, Pittsburgh PA 15213, United States of America

Received 14 June 2023; received in revised form 24 July 2023; accepted 2 August 2023
Available online xxxx

Graphical Abstract



Abstract

Finite element analysis (FEA), a common approach for simulating stress distribution for a given geometry, is generally associated with high computational cost, especially when high mesh resolution is required. Furthermore, the non-adaptive nature of FEA requires the entire model to be solved even for minor geometric variations creating a bottleneck during iterative design optimization. This necessitates a framework that can efficiently predict stress distribution in geometries based on given

* Corresponding author.
E-mail address: barati@cmu.edu (A. Barati Farimani).

https://doi.org/10.1016/j.cma.2023.116343

boundary and loading conditions. In this paper, we present StressD, a framework for predicting von Mises stress fields based on the denoising diffusion model. The StressD framework involves two models, a U-net-based denoising diffusion model and an auxiliary network to generate and predict stress distribution in structures. The denoising diffusion model generates a normalized stress map based on the given geometry, boundary conditions and loading condition, while the auxiliary network is used to determine the scaling information needed to un-normalize the generated stress map. We evaluate the StressD framework on cantilever structures and show that it is able to accurately predict von Mises stress fields while significantly reducing computational cost compared to traditional FEA.

Keywords: Deep learning; Denoising Diffusion Probabilistic model; Vision Transformer; High resolution Stress field prediction; FEA Surrogate model

1. Introduction

The emergence of new manufacturing paradigms like additive manufacturing [1–3], and digital manufacturing [4–6] as well as advances in metamaterials [7,8], combined with generative design [9,10] and topology optimization approaches [11,12], has created a demand for fast and accurate methods for predicting stress and deformation fields in structures to manufacture an optimally designed product. This task has been typically accomplished using finite element analysis (FEA) solvers that use a multi-scale approach to solve multi-physics partial differential equations (PDEs) on a discretized domain using finite element methods (FEM) [13–15]. Using iterative solvers or direct factorization-based solvers, a linear FEA model for solid mechanics solves for $F = KQ$, where F is the force vector, K is the stiffness matrix assembled from material parameters, boundary conditions, and initial conditions, and Q is the nodal displacement matrix. The material properties and the derived nodal displacement matrix are then used to calculate the nodal stress and strain tensors [16]. With an increase in geometric complexity, non-linear material characteristics, and high mesh resolution, the conventional FEA-based simulations become impractical for generative design and topology optimization, particularly when dealing with a large design parameter space.

Deep learning-based methods have enabled us to effectively process large volumes of complex data and extract meaningful features leading to the development of highly accurate models that have transformed various fields, such as fluid mechanics [17–19], material property prediction [20–22], topology optimization [23,24], among others [25]. There have been attempts to accelerate FEA analysis through the application of deep learning, including creating models for the constitutive relation of the material [26,27], improving the numerical quadrature of the FEM stiffness matrix [28], and verifying the plausibility of the FEM model using convolution neural networks as classifiers [29]. However, these methods still rely on finite element methods-based solvers, and as non-linear models become more complex with the integration of multi-physics PDEs to narrow the gap between simulations and experimental data, these finite element-based methods are hitting their limitations.

To mitigate the aforementioned issues, a viable option is to adopt a data-driven approach by developing a machine learning-based surrogate model [22,30,31] to learn the underlying phenomena from existing FEA simulation and make fast inferences on unseen samples, which would otherwise require running the entire FEA simulation again thereby reducing the computational time during topology optimization or generative design.

Several deep learning-based surrogate models have been proposed, such as Physics-Informed Neural Networks (PINNs) and DeepONets [32], which have been applied in solid mechanics as surrogate models [33–36] to identify material properties [37], predict stress distribution for plastic deformation [38,38,39], elastoplastic structures [40], as well as to detect internal structures and defects [41]. While these models have demonstrated remarkable accuracy, these models are primarily dependent on sampling points in the material domain [42] and require prior knowledge of the system to model the loss function, thereby limiting their applicability for complex geometries and multi-physics systems.

Given the significance of physical fields like distributions of strain or stress tensors in various design applications and engineering analyses, various convolution neural network (CNN) based deep learning models have been proposed such as difference-based deep learning framework [43] for predicting stress in composites with random volume fractions, bayesian based neural network to predict stress fields in porous structures [44], ResNet [45] based network for predicting stress in 2D cantilever structures [46]. In addition to this, U-Net based networks have been proposed for predicting structural response of defect-containing AM microstructures [47], stress in inhomogeneous non-linear materials [48], and stress field prediction in fiber-reinforced composite materials [49]. Most recently, multi-headed self-attention based approaches have proven successful in predicting physical field data like stress, energy, and displacement fields, as well as overall material properties characterizing the nature of stress distributions due to applied loads and crack defects [50,51].

Generative models like Generative Adversarial Networks (GANs) [52,53], comprising two neural networks – a generator that creates new data from Gaussian noise, and a discriminator that assesses these generated samples against the training data – are trained concurrently to reach a Nash equilibrium, where neither the generator nor the discriminator can enhance their performance without the other modifying its strategy. These models have been suggested as a method of unsupervised learning that enables the creation of new data consistent with the distribution of the training data. Conditional generative adversarial networks (cGANs) [54], an extension of GANs that allow for a controlled generation by passing auxiliary information such as labels, have demonstrated superior performance compared to prediction-based networks when used on unseen data with a limited number of training samples and data with finer mesh size [46,55]. These networks have been used to predict stress and strain fields as well as material properties from material microstructure [56], predict full stress fields for heterogeneous material [57], and have also been used for the prediction of thermal stress [58].

While GAN-based networks present a range of advantages over predictive models – including their capacity to make generalized predictions on smaller and diverse datasets [59], and their ability to model complex conditional distributions – they also possess certain limitations, such as the risk of 'mode collapse', a situation where the generator fails to capture the full range of diversity present in the training data. Additionally, these networks might also face the issue of 'non-convergence', where the iterative process of learning fails to reach Nash equilibrium, causing instability in the model's performance.

To address these limitations of GANs, diffusion-based models [60] have been proposed, where Gaussian noise is incrementally added to the training sample until it becomes pure noise, allowing the neural network to progressively remove noise and learn from it [61,62]. This approach has shown improved performance compared to GAN-based networks [63]. Diffusion-based models have demonstrated exceptional performance not just for unconditional image generation [64,65] and conditional image generation [66,67], but also for a diverse range of image-based tasks, including flow field reconstruction using super-resolution [68], in painting and image-to-image translation [69].

In the realm of mechanics, materials modeling and engineering design, denoising diffusion-based models have been used to assess dynamic fracture mechanisms for complex geometries [70], predict high-resolution stress fields from material microstructures [71]. The DDPM-based approach has the potential to generate a range of possible solutions, thereby enabling the quantification of uncertainty. Owing to this capability and their ability to generate diverse, high-quality data, DDPMs are seeing increased use in addressing inverse design problems, where specific geometries are generated given certain conditions. This includes their application in the design of microstructures with fine-tuned nonlinear material properties [72], creation of microstructures for organic solar cells [73] and to generate designs tailored to a specific mechanical response [74]. These applications highlight the potential of DDPMs to offer innovative and practical solutions for inverse design problems, which typically presents a variety of potential solutions.

Diffusion-based models operate on bounded data, where all the samples within a dataset have defined bounds, such as images where every pixel's RGB value falls within the range of 0–255. However, for prediction tasks, the samples are unbounded and can potentially have any value. Therefore, to the best of our knowledge, there have been no attempts to use diffusion-based models for making predictions on such types of data. Simply scaling the entire dataset based on the values of the entire dataset can potentially hinder the generalizability of the model during testing. In addition, since diffusion-based models learn to estimate the noise added at each time step, adding a fixed amount of noise based on the entire dataset leads to some samples degrading faster than others, creating instability during the training. An alternative solution to the unboundedness of the data in diffusion-based models is to use an autoencoder to encode the inputs, as in latent diffusion models [75]. However, the accuracy of the model in such cases is heavily dependent on the ability of the decoder to reconstruct the results, introducing an additional potential source of error during prediction, especially in the case of out-of-distribution data affecting the generalizability of the model.

The current study introduces a novel conditional DDPM-based framework that predicts 2D stress distribution of a structure given geometry, boundary conditions, and loading conditions. As DDPM models operate on normalized data within the range of $[-1, 1]$, our approach involves a two-step process. Fig. 1 shows the overall schematic of

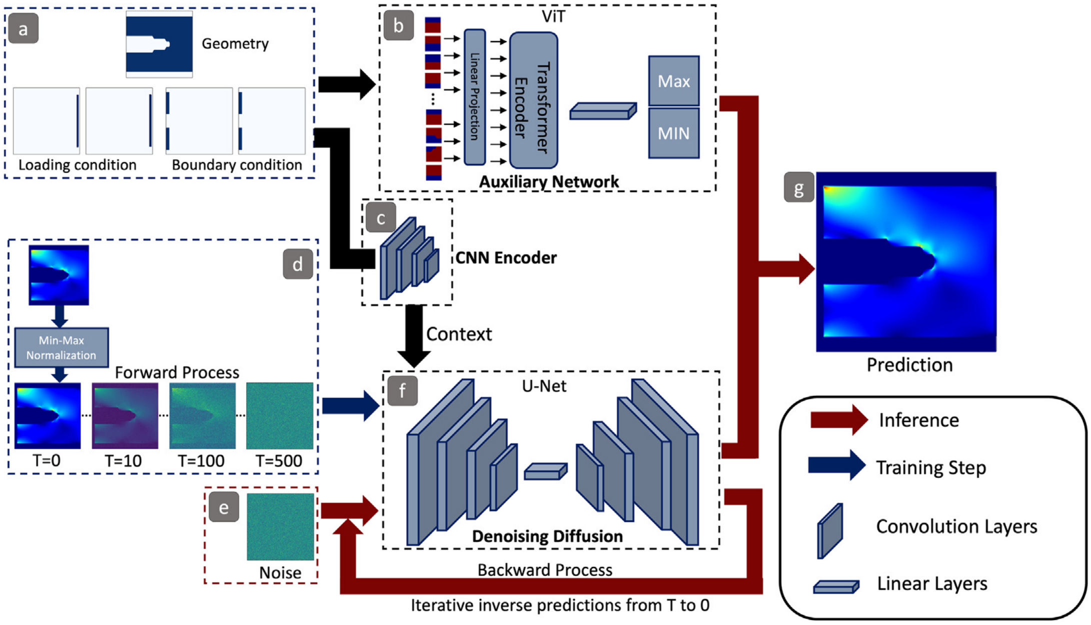


Fig. 1. Description of workflow.
(a) Inputs: geometry, boundary & loading conditions; (b) vision transformer-based auxiliary network; (c) CNN encoder creates context vector for (f) U-Net diffusion model; (d) adds Gaussian noise to ground truth stress fields; training learns noise at each step T; inference iteratively removes noise from (e) pure Gaussian noise using context vector, generating normalized stress fields; (g) predicted min/max values un-normalize generated stress fields.

our framework. First, we use DDPM to predict a normalized stress field, and then by using a vision transformer to predict the maximum and minimum values of stress, we un-normalizing and rescale the generated stress fields. We assess our framework using the dataset introduced in [46,55], demonstrating a considerable enhancement in outcomes when compared. Additionally, we evaluate our framework on both fine mesh and sparse datasets.

2. Methodology

2.1. Auxiliary network

The stress map produced by the diffusion-based method is normalized to the range of $[-1, 1]$ due to the input data being normalized in the same range. In order to obtain the stress map with actual stress values, it is necessary to perform a rescaling operation on the generated stress map. To achieve this, we employ an auxiliary network that predicts the minimum and maximum stress values required to unnormalize the normalized stress map.

Transformers, based on self-attention mechanisms, have become a popular model architecture for NLP tasks [76]. In recent years, there has been a surge of interest in applying transformer-based approaches to various computer vision applications, including object detection [77], segmentation [78], and more, highlighting a growing desire to explore the potential of these models in the field of computer vision [79]. In our work, we use the Vision Transformer (ViT) [80] as a backbone model to predict the minimum and maximum stress values required to unnormalize the stress map generated through the diffusion-based method.

Fig. 2. shows the schematic of the auxiliary network with ViT as the backbone. Here, the input data, which includes geometry, boundaries, and loading conditions, is first split into fixed-sized non-overlapping patches, which are then flattened and embedded with positional encoding to produce a sequence of embeddings. This embedding sequence is then fed into a Transformer-based encoder, which has a series of transformer blocks that use the self-attention mechanism [81]. As stress values scale with the applied force, the minimum and maximum values of force in the X and Y directions are extracted and passed through a linear layer. The output from the linear layer is combined with the output from the transformer and passed through a bilinear layer that computes the element-wise

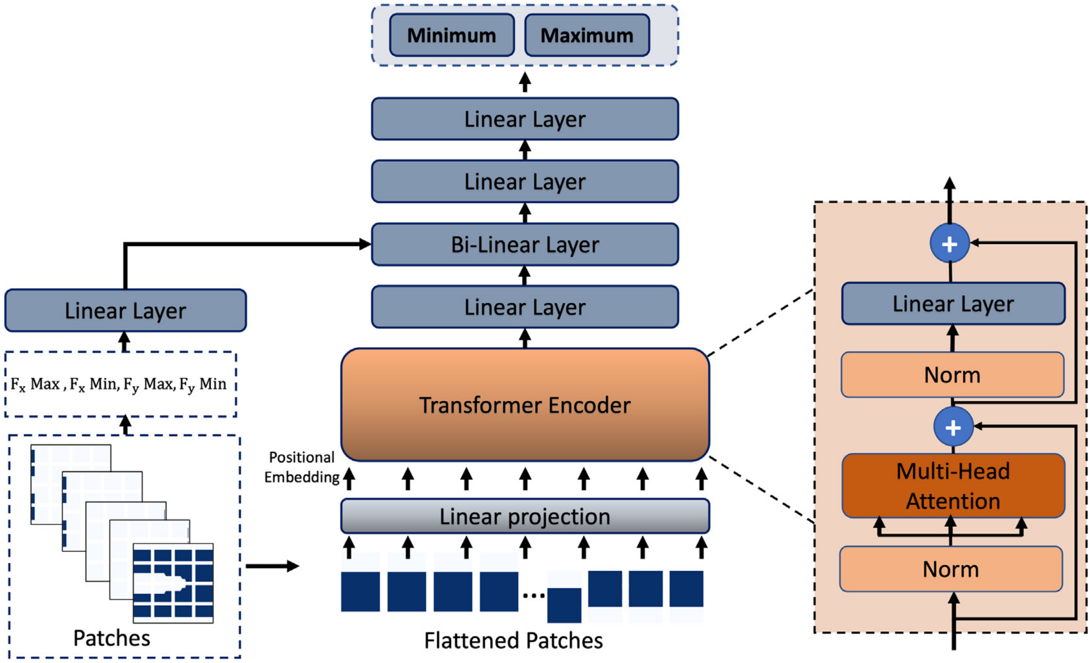


Fig. 2. Schematic of Auxiliary network with ViT backbone.
The geometry, boundary conditions, and loading conditions are split into patches of a fixed size, which are then passed through linear projection and positional embedding before being input into the transformer. The loading condition is further processed to extract the maximum and minimum values, which are passed through a linear layer. The resulting output from the linear layer and that of the transformer are subsequently passed through a bilinear layer, followed by a series of linear layers to obtain the minimum and maximum stress values.

product of the feature from these two inputs, followed by a series of linear layers to predict the minimum and maximum stress.

2.2. Denoising diffusion probabilistic model (DDPM)

Denoising diffusion probabilistic models (DDPM) [61] involves defining a Markov chain of diffusion steps to gradually add tractable noise to the input data in the forward process. Considering normalized stress map as input x_0, the forward diffusion process can be modeled as a Markov chain [61] :

$$\begin{aligned} &q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \\ &q(x_{1:T}|x_0) = \textstyle\prod_{t=1}^{T} q(x_t|x_{t-1}) \end{aligned} \tag{1}$$

Where $\beta_t \in (0, 1)$ is the variance scheduler which controls the step size of noise added.

Our goal is to generate meaningful data from Gaussian noise input $x_T \approx \mathcal{N}(0, I)$ given a conditioning vector 'z'. The generation process (backward process) is the reverse of the forward process where the neural network p, learns to recover the x_{t-1} given x_t and z values (see Fig. 3). The backward process can be written as:

$$p(x_0 : T) = p(x_T) \prod_{t=1}^{T} p(x_{t-1}|x_t, z) \tag{2}$$

2.3. Datasets

To assess the performance of our approach, we employed four datasets, namely a coarse mesh dataset, a fine mesh dataset, a sparse dataset and hyper elastic dataset, as shown in Fig. 4.

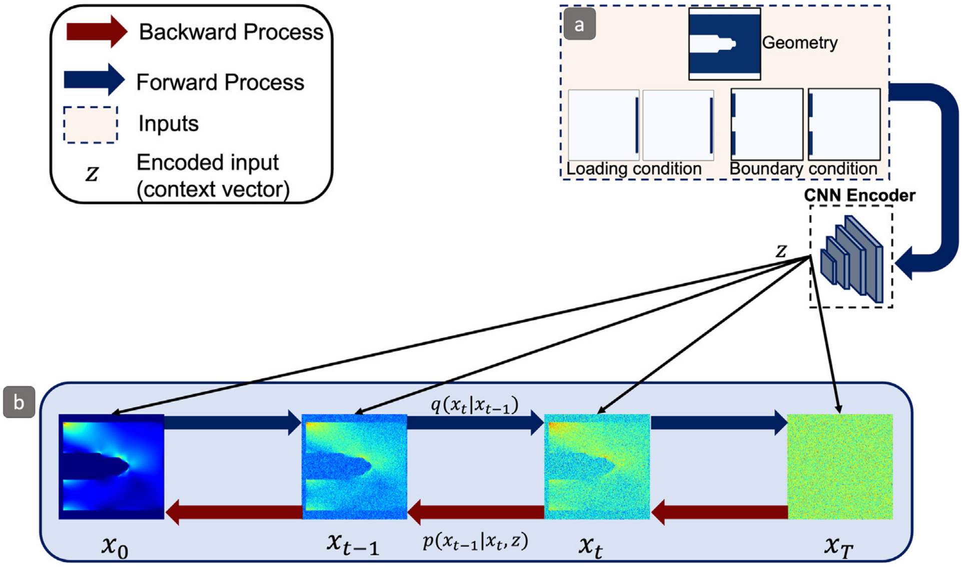


Fig. 3. Schematic for training DDPM.
The input consists of geometry, boundary conditions, and loading conditions, is first fed through a convolutional neural network to generate a context vector z for conditioning the Denoising Diffusion Probabilistic Model (DDPM). In the forward process, the normalized stress map x is subjected to tractable noise for T timesteps, gradually becoming more degraded, at $t = T$ resembling pure Gaussian noise. During the backward process, the model learns to estimate the added noise at each timestep t given the conditioning z and x_t.

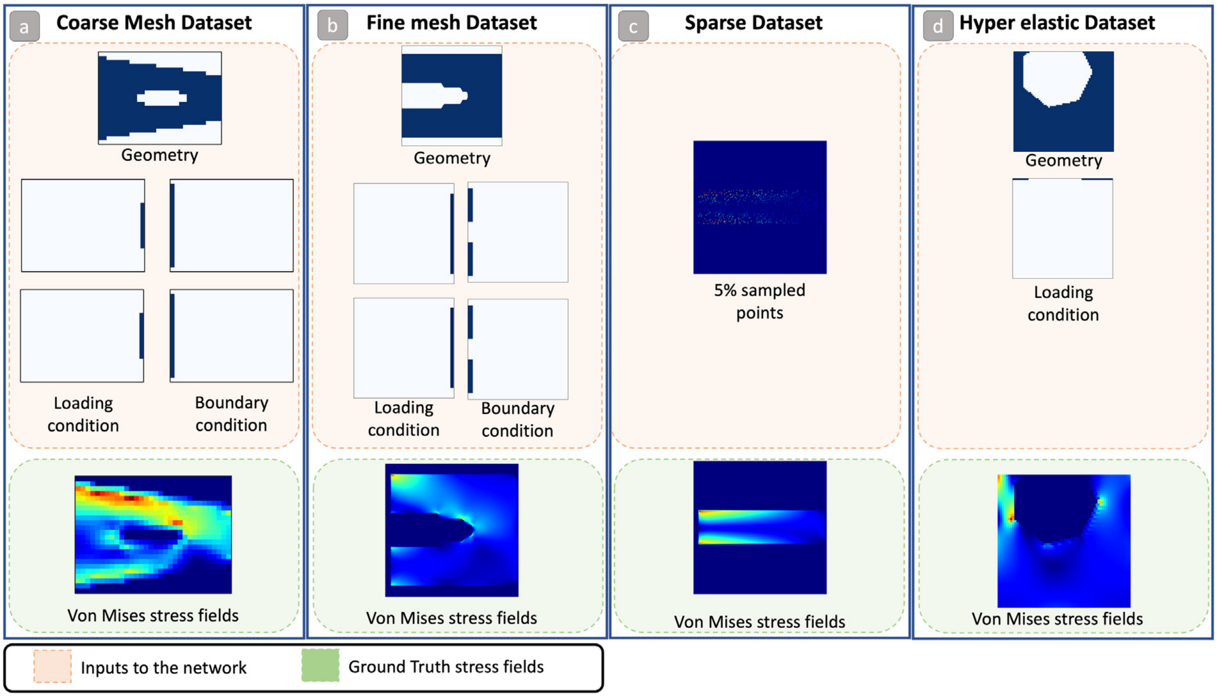


Fig. 4. Datasets used for evaluation.
(a) The coarse mesh dataset with mesh size 24 × 32. (b) Fine mesh dataset of mesh size 256 × 256 and (c) Sparse dataset with 5% uniformly sampled points in the material domain.

2.3.1. Coarse mesh dataset

In our analysis, we utilized Nie et al.'s multichannel dataset [46,55], which considers a cantilever structure composed of homogeneous and isotropic linear elastic material, with the left end of the structure fixed and the right end bearing evenly distributed external static load in both horizontal and vertical directions.

The dataset contains 80 geometries divided into six groups, and has a mesh size of 32 × 24. Each datapoint comprises a total of 5 channels. Each geometry (channel 1) is defined by a binary array, where the number 1 signifies the material domain and 0 signifies an empty space. The two boundary condition channels denote the fixed left end, attributed with a value of -1 at the location where it is fixed. The right end of the structure, where the load is applied, is signified in the two load condition channels, indicating the force's magnitude in the x and y directions. The load orientation varies from 0 degrees to 355 degrees in increments of 5 degrees, and for each orientation, the load magnitude increases from 0N to 100N in increments of 5N. Finally, the fifth channel contains the von Mises stress field. This dataset comprises a total of 120,960 cases. Fig. 4(a) provides a visual representation of the geometry, boundary conditions, and loading conditions that are used as input to our model, along with the von Mises stress fields, which is the output we are aiming to predict.

2.3.2. Fine mesh dataset

To produce the fine mesh dataset, we uniformly sampled 50,000 data points from the coarse mesh dataset. We then upsampled the geometry channel to a mesh size of 256 × 256, maintaining the boundary and loading conditions. To compute the stress fields, we assumed that all elements within the material domain were 4-node quadrilaterals with a size of 1 mm × 1 mm, and solved them using SolidPy [82] FEM solver. Each problem takes around 45 s to solve.

2.3.3. Sparse dataset

In order to simulate sparse sensor placement for Structural Health Monitoring (SHM) [83–85] with a focus on preventive maintenance, we incorporated the fine mesh dataset in our training approach. This involved randomly sampling 5% of points in the von Mises stress fields within the material domain during training, with the sampled points being different every time the data point was encountered in subsequent epochs. The main objective of the task was to complete the sparse stress field, with no information being provided about the geometry, loading condition, or boundary condition. This problem can be perceived as similar to a task resembling inpainting.

2.3.4. Hyper-elastic dataset

To evaluate both the performance and the underlying learning capabilities of our model when applied to non-linear constitutive models, we created a distinctive dataset based on a 2D hyperelastic problem. This specific problem was selected due to its deviation from the linear elastic model that was employed in our previous datasets.

Hyperelasticity, as a non-linear problem, necessitates a more complex understanding of material behavior, especially when large deformations are involved, as opposed to relatively straightforward linear-elastic relations. This provides a more challenging, yet realistic, test case for our model, allowing us to evaluate its adaptability and learning prowess in a more demanding scenario. This shift from a linear to a non-linear problem thus offers an interesting and robust assessment of our model's ability to grasp and accurately predict complex material behavior.

To generate a comprehensive and varied dataset, we constructed 49,750 unique datapoints, each with a distinct set of parameters. These parameters include the number of sides (ranging from 3 to 7) of a polygonal cut introduced into a square plate, the position and dimensions of the cut, and the magnitude of force applied. In these configurations, the square plate was fixed at one end while the force was applied at the opposite end.

For the numerical solutions to these problems, we utilized FEniCS [86–89], an open-source platform for solving partial differential equations (PDEs). To ensure compatibility with our model, we converted the resulting solution fields (which were originally obtained on a triangular mesh) to a structured 2D rectangular grid. This step involved the interpolation of stress field data from the FEniCS solution to a 2D numpy array using PyVista. The detailed problem formulation is provided in Appendix:A.

The training phase of our model utilized 39,800 data points, each representative of unique problem settings, and the remaining 9,950 datapoints were used as test set for evaluation.

In order to rigorously assess our model's capabilities of generating and predicting results for unseen geometries, we developed an additional test set comprising 10,000 datapoints. These datapoints were characterized by an octagonal (8-sided) cut in the square plate, varying in dimensions and positions. Notably, this specific geometric configuration was not part of the model's training data, ensuring a stringent test of the model's ability to generalize beyond its training scope.

2.4. Implementation details

Our DDPM network adopts the U-Net architecture [90]. Initially, we input the four channels of geometry, loading, and boundary conditions into a CNN-based encoder, which maps them to a one-channel array that represents the context z. We then normalize the von Mises stress fields using min–max normalization to scale the data between -1 and 1 and add noise to this normalized stress map using a cosine variance scheduler for 500 time steps. The context z and the noised stress map are combined to form a two-channel image, which we feed into a conditional U-Net that comprises ReNet blocks [45]. The U-Net incorporates residual connections between its upsampling and downsampling layers. We embed the time step t representing the magnitude of the added noise by utilizing sinusoidal position embeddings [76]. Moreover, we flatten the context vector after it undergoes a series of convolution layers and add it to the time embeddings for better guidance. Adam optimizer [91] is utilized to optimize the network, with the objective of minimizing mean absolute error (L1 Loss). Additionally, we implement a variable learning rate that is reduced on plateau. In case of a sparse dataset, we input a sparse stress map with only one channel, and the process remains the same.

The input to the auxiliary network includes the geometry condition, loading condition, and boundary condition for non-sparse datasets, and only sparse data for sparse datasets. This input is presented in the form of a 4-channel array, which is partitioned into patches of size 4 for coarse mesh datasets and 16 for fine mesh datasets and sparse data. These patches are then converted into a linear embedding of 256 and then fed into the transformer blocks to predict the minimum and maximum values. Detailed implementation details are provided in SI.

3. Results

To compare the performance of our method with existing techniques such as StressGAN [55] and StressNet [46], we adopted a range of metrics such as mean absolute error, mean squared error, peak stress absolute error, and percentage peak stress absolute error (MAPE). However, our network was specifically designed to minimize the mean absolute error.

Given predictions $\hat{y}$ and ground truth values as y we define our metrics as:

$$MAE = \frac{1}{n}\sum_{j=1}^{n} \| y_j - \hat{y}_j \|$$

$$MSE = \frac{1}{n}\sum_{j=1}^{n} (y_j - \hat{y}_j)^2$$

$$MAPE = \frac{MAE}{max\{y\} - min\{y\}} * 100\%$$

$$PAE = \| max\{y\} - max\{\hat{y}\} \|$$

To assess the performance of our methods on coarse and fine mesh datasets, we randomly partitioned each dataset into training and testing sets. Specifically, we used 80% of the total coarse mesh dataset (100,000 samples) for training and the remaining 20% (20,000 samples) for testing. Similarly, for the fine mesh dataset, we trained our model on 40,000 samples and evaluated it on 9,100 data samples. Table 1 shows the quantitative evaluation of our approach, it is observed that our network shows significant improvement compared to existing models. Additionally, our evaluation on the hyperelastic dataset shows that our framework's performance is consistent with the linear elastic dataset. This reinforces the premise that our data-driven approach, which does not incorporate any physics-based or model-based loss in its loss function, is agnostic to the underlying finite element model. This suggests that our approach should yield similar results across different FEM models, provided that sufficient data is made available to ensure its generalizability. Fig. 5 shows the stress map predicted by the StressD framework for a random sample from the test set, highlighting its strong alignment with the ground truth.

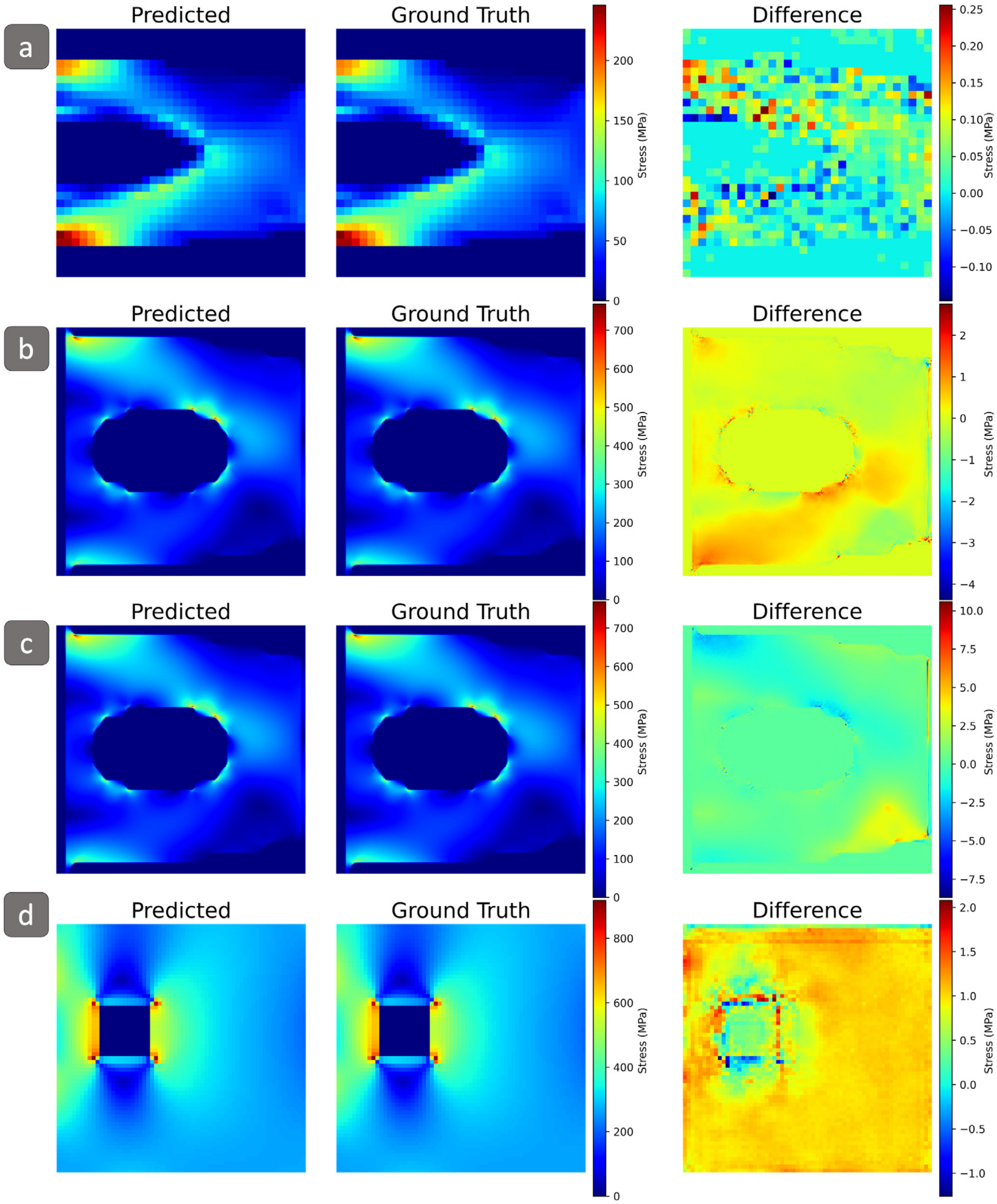


Fig. 5. Predictions made by StressD on test dataset.
Prediction, ground truth and difference for the test set of (a) Coarse mesh dataset, (b) Fine mesh dataset, (c) Sparse dataset, and (d) hyper-elastic dataset.

3.1. Sampling time-steps and MAE

The quality of the samples generated from a diffusion model during inference is heavily influenced by the number of time steps used in the iterative diffusion process. While increasing the number of time steps can result in more accurate and precise samples, it can also lead to longer inference times due to the iterative nature of the

Table 1
Quantitative evaluation of StressD framework.

Network	Dataset	MAE	MSE	MAPE	PAE
StressGAN	Coarse Mesh dataset	0.19	0.13	0.14%	0.48
StressNet		0.2	0.15	0.15%	0.5
StressD (ours)		**0.0418**	**0.0125**	**0.02%**	**0.0327**
	Fine Mesh dataset	**0.4889**	**23.0691**	**0.13%**	**1.5595**
	Sparse Dataset	**1.1248**	**7.3714**	-	**7.0718**
	Hyper-elastic dataset	**1.5818**	**234.8559**	**0.05%**	**10.1819**

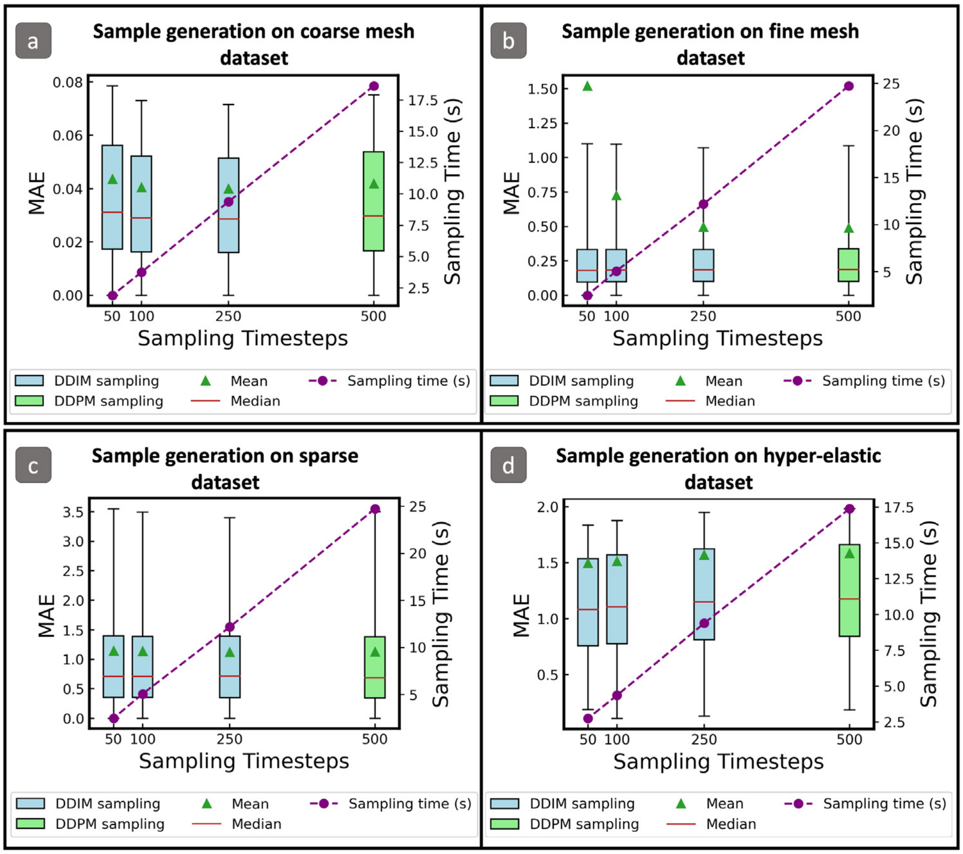


Fig. 6. Sample generation and related MAE values for different sampling time steps.
Time taken to generate samples using DDIM and DDPM for different time steps on a (a) coarse mesh dataset and (b) fine mesh dataset (c) Sparse dataset and (d) Hyperelastic dataset.

sampling process. To address this issue, recent research [62,66,92] has been focused on enhancing the speed of the sampling process while simultaneously improving the quality of the generated samples, one such approach, Denoising Diffusion Implicit Model or DDIM, [93] exploits the fact that the minimizing objective depends only on $q(x_t|x_0)$ and not on $q(x_{1:T}|x_0)$ as formulated in DDPM, to make an inference process on fewer steps while maintaining the quality of generated samples, using an existing model trained on DDPM objective.

Fig. 6 shows a comparison between DDPM and DDIM in terms of sampling time and Mean Absolute Error (MAE) when generating samples from a test dataset. The box plot illustrates the 0–95 percentile range of the data, including mean and median values. The sampling time comparison was conducted using a batch size of 1 on GPU.

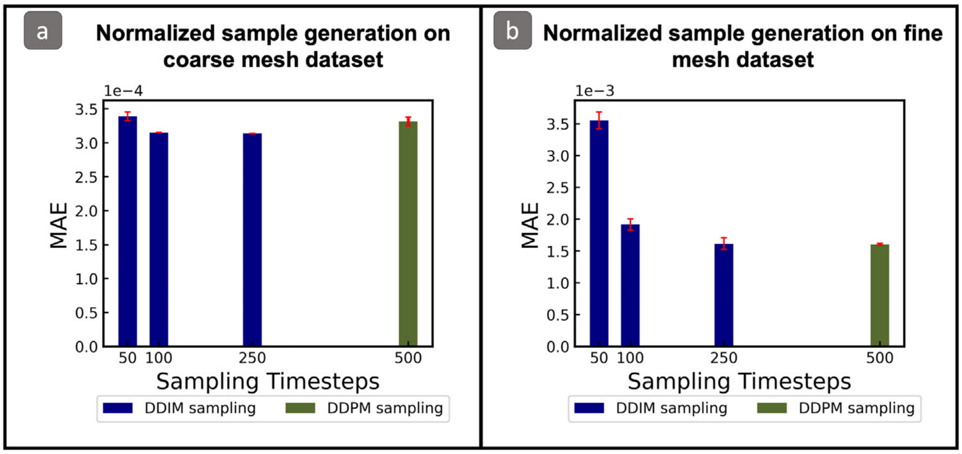


Fig. 7. Diffusion model performance.
Performance of denoising diffusion model with DDPM and DDIM sampling on (a) coarse mesh dataset and (b) fine mesh dataset.

Since denoising diffusion-based models are stochastic and samples are generated from Gaussian noise, the figure shows the average MAE values obtained from three complete evaluations of the test dataset for each method.

According to the findings, sampling with DDIM for 250-time steps results in faster sample generation compared to DDPM. Moreover, DDIM either performs better or equally well as DDPM in terms of mean absolute error values for this time step. However, as the number of sampling time steps decreases, DDIM's MAE performance decreases, and the generated samples have greater variations.

3.2. Performance of individual models

As our method does not entail any learning during the scaling process and the two networks operate independently, it is essential to evaluate the performance of each network separately. This is due to the potential for error amplification in the generated samples, contingent on the maximum stress value predicted by the ViT network. Fig. 7 provides a performance snapshot of the denoising diffusion-based model's capability in generating normalized stress fields. Given the stochastic nature of these models, we report the MAE values averaged across three runs. For the coarse mesh dataset, the MAE value of the ViT network in predicting the maximum and minimum values, essential for stress field normalization, stands at 0.01 MPa. Conversely, for the fine mesh dataset, the MAE value is marginally higher at 0.48 MPa.

3.3. Generalization on unseen geometry

Using the model trained on the hyperelastic dataset, we performed tests on an unseen geometry - a dataset of 10,000 datapoints featuring a square plate with an octagonal cut with similar formulation as hyperelastic dataset. This geometry was intentionally excluded from the training data to rigorously assess the model's ability for generalization. With a MAE of 133 MPa and a Percentage Peak Stress Absolute Error (MAPE) of 1.7%, our model has demonstrated robust performance in terms of generalizability, as evidenced by its performance on unseen geometries. This highlights its reliability in predicting stress distributions across diverse conditions (see Fig. 8).

Furthermore, it is worth noting as shown in Fig. 9, that the diffusion model plays a significant role in accurately generalizing the stress distribution for unseen geometry. It was also found that the auxiliary network, which was designed to predict the inverse scaling value, was a major contributor to the overall prediction error.

4. Discussion

4.1. Model performance

We have presented a deep learning approach for predicting von Mises stress fields for linear as well as non-linear constitutive models by merging a denoising diffusion probabilistic model with a vision transformer. By incorporating

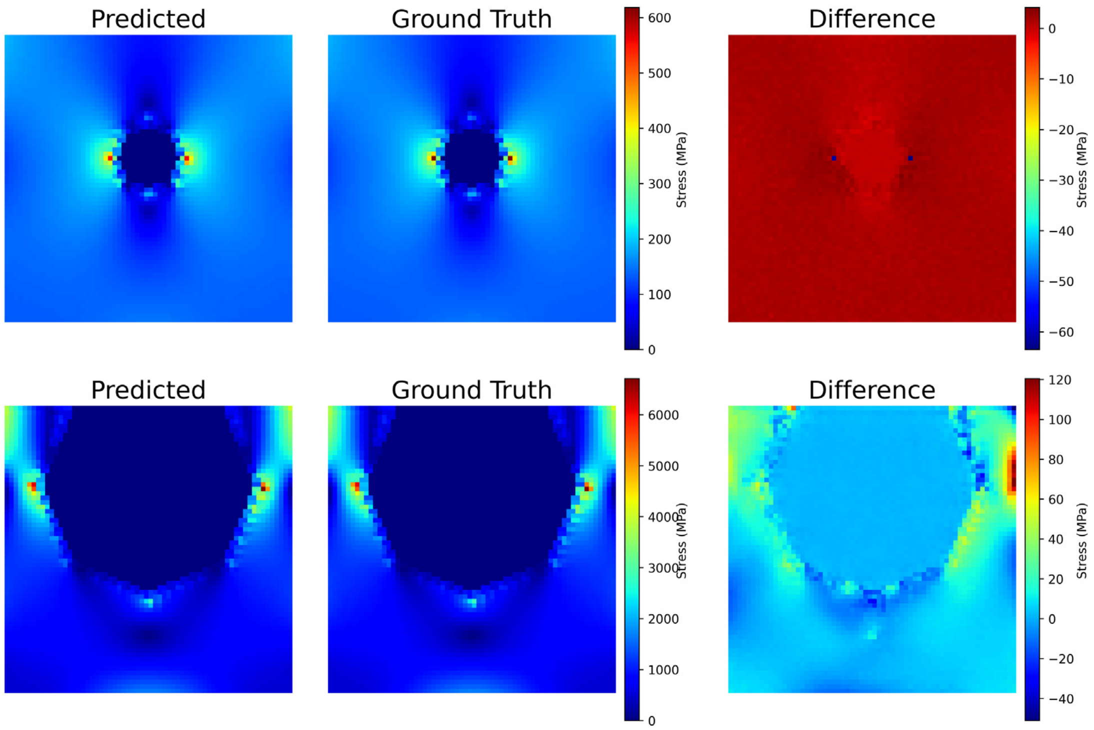


Fig. 8. Predictions made by StressD framework on unseen geometry from hyperelastic dataset.
Prediction, ground truth and difference of randomly selected samples generated from StressD framework on unseen geometry for hyperelastic problem.

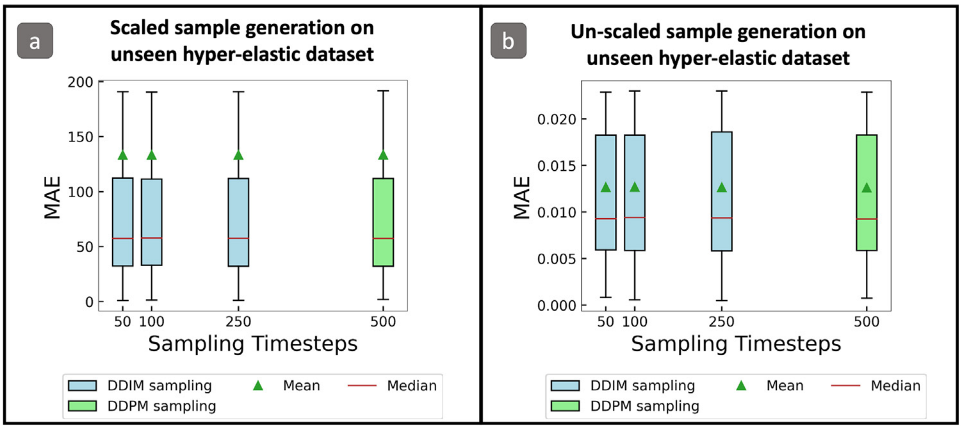


Fig. 9. Assessment of Model Performance on Unseen Geometries with the Hyperelastic condition.
Evaluation of the Mean Absolute Error (MAE) metric for (a) the StressD framework applied to the unseen geometry dataset under hyperelastic conditions, and (b) the Diffusion model.

the DDPM model with an auxiliary network (in our case, a vision transformer), we have effectively divided the stress field prediction task into generation and scale prediction tasks. This enables us to utilize DDPM-based models, which have been employed as generative networks, along with vision transformers, which have primarily been used for class prediction.

The performance metrics of our model speak volumes about its effectiveness. We observed an approximate 78% improvement in MAE (Mean Absolute Error) results and a significant 92% enhancement in MSE (Mean

Squared Error) values compared to previous models. Furthermore, our model's capability to accurately predict non-linear hyperelastic problems without necessitating substantial changes in the network architecture demonstrates its versatility. This suggests that our proposed approach is not bound by the specificities of a finite element method (FEM) model, but rather it can be universally applied across various problem types, reflecting its broad applicability and utility.

4.2. Efficiency and speed

Our approach provides a substantial enhancement in computation speed while maintaining accuracy, ensuring no significant compromise on accuracy. A comparative analysis with FEM revealed that our DDPM-based model facilitates nearly a 44.4% speedup in sample generation time, generating one sample in around 25 s as opposed to the typical 45 s required by FEM on the fine mesh dataset (256 × 256). This improvement in time for results generated from our model is even more pronounced when utilizing the DDIM sampling approach, which allows the inference time to be reduced to an impressive 2 s for 50 time steps. Additionally, in the case of the non-linear hyperelastic dataset (size 64 × 64), the inference time for 500 time steps is nearly 17.5 s. Employing the DDIM approach for 50 time-steps shrinks this to approximately under 2 s, achieving slightly better performance without a considerable uptick in error. In the case of the coarse mesh dataset the inference time for 500 time steps is around 18 s and for 50 time steps, using DDIM sampling, is under 1 s.

This considerable reduction in inference time does not lead to a significant compromise in prediction accuracy, thereby achieving an optimal balance between speed and precision. In contrast to FEM, which experiences an exponential increase in solution time with rising mesh size and precision requirements, our approach demonstrates a constant and much faster alternative due to its reliance on matrix multiplication for a set number of times rather than iterative optimization regardless of the underlying constitutive and FEM model.

4.3. Selection and rationale for the neural network model

We employed diffusion-based models that progressively adds tractable noise into the data during training until the data completely deteriorated and resembles Gaussian noise. The neural network is trained to predict the incrementally added noise at each time-step, given the noised data, the conditioning vector, and the corresponding time-step. During the reverse process, the network guides an organized removal of the noise until the original data is reconstructed. In doing so, it effectively learns both the high and low-frequency features of the data. In the context of our stress prediction model, this approach facilitates learning underlying patterns in the data amidst the uncertainty introduced by the added noise. The noise functions as a form of regularization, boosting the performance of the model and enhancing its ability to generalize. This becomes particularly advantageous when dealing with complex, high-dimensional data where the model's robustness to minor variations and noise is paramount.

One significant advantage of using transformers is their self-attention mechanism, which has a global receptive field, as opposed to the local receptive fields of CNNs. This means that transformers can identify dependencies across an entire image, not just local areas. Additionally, transformers incorporate positional information about pixels, proving particularly valuable when dealing with sparse datasets where only 5% of points within the material domain are sampled. Transformers' adeptness in predicting two values (minimum and maximum) was found superior, especially for sparse datasets. Initial experiments with CNNs for similar predictions yielded less satisfactory results, potentially due to the inherent 'averaging out' of values within the CNN's kernel window.

4.4. Scaling of values

In our model, scaling serves a crucial function by normalizing the stress fields, thus facilitating their generation by the DDPM. Following this, the Vision Transformer (ViT) predicts the necessary scaling factors, specifically the minimum and maximum values of the stress under given geometry, loading conditions, and boundary conditions. This process enables the re-transformation of normalized stress fields back to their original stress magnitudes. This strategic partition of the stress prediction task into two distinct parts – stress field generation and scale generation – allows for an efficient and powerful solution for stress field prediction.

This separation of tasks leverages the respective strengths of both DDPM and ViT, providing a model that is both efficient and effective. Notably, this division of labor allows the neural network to learn more effectively, as it alleviates the requirement for the network to handle the full complexity of predicting stress maps and scaling factors simultaneously. The network can focus on learning the nuanced patterns and correlations in the stress fields without the additional complexity of varying magnitudes, thereby enhancing its overall performance. An added benefit of our approach stems from the independence of the two networks – the DDPM and the ViT – in our architecture. In more traditional, interconnected networks, error or loss in one part of the model can potentially affect other parts, leading to a domino effect that may complicate diagnosis and correction. However, by maintaining a level of independence between the DDPM and the ViT, it is easier to isolate and identify the source of any inaccuracies or inefficiencies. This division ensures that the impact of any poor prediction remains confined within its respective task—either the generation of stress maps or the determination of scaling factors.

4.5. Future work

In our study, we have applied the Denoising Diffusion Probabilistic Models (DDPMs) to 2D geometrical problems. However, real-world engineering scenarios often involve three-dimensional structures, adding a degree of complexity both in terms of problem formulation and computational requirements. The extension of our model to cater to 3D geometries remains a challenging but promising future research direction, potentially broadening the applicability of our approach.

Our current work revolves around the forward problem of computing stress distributions given geometries, loads, and boundary conditions. Yet, the concept of DDPMs opens up intriguing possibilities in tackling inverse design problems. These problems inherently revolve around determining the optimal or desired outcomes—be it the optimal distribution of materials given a set of constraints and objectives in topology optimization, or estimating material parameters or loads from observable data like displacements or stresses in parameter identification problems. The adaptation of our model to these problems could be an innovative extension of our work.

In addition, integrating structural dynamics into the predictive model could be another interesting avenue to explore. This would imply a shift from predicting static stress distributions to computing dynamic stress changes over time given a specific dynamic load. This extension, while challenging, could significantly increase the model's usefulness in more diverse engineering scenarios.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgment

Research was sponsored by the Army Research Laboratory, USA and was accomplished under Cooperative Agreement Number W911NF-20-2-0175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

We extend our gratitude to Abraham George, AmirPouya Hemmasiann, Janghoon Ock and Kazem Meidani for their invaluable feedback, which significantly contributed to the refinement of this paper.

Appendix A. Hyperelastic problem formulation

In order to assess the applicability of our model for non-linear problems we generated a dataset using hyperelastic problem formulation as provided by "Fenics solid tutorials" [94], for detailed formulation please check the reference. Considering deformation:

$$y(X) := X + u(X)$$

$$F(X) := Grady(X)$$

The length, surface area, and volume are then transformed by

$$dl = FdL, \quad ds = (detF)F^{-T}dS, \quad dv = detFdV$$

This implies that the unit normal N in the reference configuration has a corresponding relationship with the unit normal in the deformed configuration.

$$n = \frac{det(F)F^{-T}N}{\|det(F)F^{-T}N\|}$$

When using the Lagrangian description, it is more appropriate to use the 1st Piola–Kirchhoff stress tensor P. This tensor maps a normal N to its corresponding traction in the deformed configuration. This tensor has a specific relationship to the Cauchy stress tensor σ, and rewriting it in terms of body forces B and surface forces G:

$$P = (detF)\sigma F^{-T}$$

$$B = (detF)b$$

$$G = \|(detF)f^{-T}N\|g$$

Rewriting in Lagrangian coordinates:

$$\operatorname{Div}\mathbf{P} + \mathbf{B} = \mathbf{0} \qquad \text{in } \Omega, \tag{A.1}$$

$$\mathbf{P}\cdot\mathbf{N} = \mathbf{G}, \qquad \text{on } \Gamma_N, \tag{A.2}$$

$$\mathbf{u} = \bar{\mathbf{u}}, \qquad \text{on } \Gamma_D, \tag{A.3}$$

This is similar to Euler–Lagrange equations of the energy functional minimized over the displacements satisfying the dirichlet boundary condition. The 1st Piola–Kirchhoff stress tensor is then given by the stored energy density W by

$$I(\mathbf{u}) = \int_\Omega W(\mathbf{F})\,\mathrm{d}V - \int_\Omega \mathbf{B}\cdot\mathbf{u}\,\mathrm{d}V - \int_{\partial\Omega} \mathbf{G}\cdot\mathbf{u}\,\mathrm{d}S \tag{A.4}$$

$$\mathbf{P} = \frac{\partial W}{\partial \mathbf{F}}. \tag{A.5}$$

Here we consider the Mooney–Rivlin energy:

$$W(\mathbf{F}) = a|\mathbf{F}|^2 + b|J\mathbf{F}^{-\top}|^2 + cJ^2 - d\ln J + e, \qquad J = \det\mathbf{F}.$$

The positive parameters (except of e) can be expressed in terms of the Lame constants λ and μ. Here we have used a simplified form called Neo-Hook energy:

$$W(\mathbf{F}) = \frac{\mu}{2}(|\mathbf{F}|^2 - 3 - 2\ln\det\mathbf{F})$$

Appendix B. Transformer and self-attention

The self-attention mechanism is a driving factor of the transformer model [76,95], designed to discern long-range dependencies and contextual nuances within input data. In Vision Transformer (ViT) [80] model, this mechanism identifies various regions of the input in the context of the whole image, evaluating their significance in relation to the task being executed.

The self-attention mechanism creates a weighted sum of the input data. The weighting factors are derived from the level of similarity between the input features. This methodology enables the model to emphasize the input features that hold more relevance, facilitating the extraction of more informative representations of the input data.

At its core, the self-attention component operates as a computational unit that measures the interactions between pairwise entities. It enables a network to understand and learn the underlying hierarchies and alignments within the input data. As a result, it has emerged as a critical factor in enhancing the robustness of visiozzn networks. In our work, we employed the Multi-Head Scaled Dot Product Attention mechanism as proposed by Vaswani et al. [76]. Unlike the single-headed attention, the multi-head attention mechanism operates by computing the attention multiple times in parallel. Each individual 'head' computes an independent attention output, and these outputs are then concatenated and linearly transformed to match the expected dimensions.

The Multi-Head Scaled Dot Product Attention can be mathematically represented as [95]:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O$$

where each individual head (n) is computed as:

$$\text{head}_i = \text{Attention}(QW_{Qi}, KW_{Ki}, VW_{Vi})$$

and Attention is computed using the Scaled Dot-Product Attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{n}}\right)V$$

In the above formula K, Q and V are key, query and value respectively and $QW_{Qi}, KW_{Ki}, VW_{Vi}$ are the parameters.

Appendix C. System information

All models were trained on 4 NVIDIA GeForce RTX 2080 Ti using data distributed parallel module provided by PyTorch. All inferences were done on NVIDIA GeForce RTX 1080. FEM data generation was done on CPU Intel(R) Core(TM) i9-9900K CPU @ 3.60 GHz using a single core.

Appendix D. Model information

Dataset	DDPM (U-Net)	Number of parameters	ViT	Number of parameters
Fine Mesh dataset	Dimension= 256 × 256 Input dimension: 5 Output dimension: 1 ResNet layers stages and blocks: 1,1,2,2,4,4	**38,065,023**	Embedding Dimension=256 Hidden dimension= 512 Number of heads = 8 Number of layers =18 Patch size = 16	**12,121,450**
Sparse Mesh dataset	Dimension= 256 × 256 Input dimension: 1 Output dimension: 1 ResNet layers stages and blocks: 1,1,2,2,4,4	**38,052,379**	Embedding Dimension=256 Hidden dimension= 512 Number of heads = 8 Number of layers =18 Patch size = 16	**11,859,050**
Coarse Mesh dataset	Dimension= 32 × 32 Input dimension: 5 Output dimension: 1 ResNet layers stages and blocks: 1,1,2,2,4	**30,579,583**	Embedding Dimension=256 Hidden dimension= 1024 Number of heads = 8 Number of layers =12 Patch size = 4	**11,754,346**
Hyperelastic dataset	Dimension= 64 × 64 Input dimension: 2 Output dimension: 1 ResNet layers stages and blocks: 1,1,2,2,4	**30,901,822**	Embedding Dimension=256 Hidden dimension= 512 Number of heads = 8 Number of layers =16 Patch size = 4	**10,747,754**

Appendix E. Data availability

Please find the relevant code for FEM data generation and network: https://github.com/BaratiLab/StressD
Data can be shared upon request.

References

[1] Y. Lakhdar, C. Tuck, J. Binner, A. Terry, R. Goodridge, Additive manufacturing of advanced ceramic materials, Prog. Mater. Sci. 116 (2021) 100736.

[2] B. Blakey-Milner, P. Gradl, G. Snedden, M. Brooks, J. Pitot, E. Lopez, M. Leary, F. Berto, A. du Plessis, Metal additive manufacturing in aerospace: A review, Mater. Des. 209 (2021) 110008.

[3] M.B. Kumar, P. Sathiya, Methods and materials for additive manufacturing: A critical review on advancements and challenges, Thin-Walled Struct. 159 (2021) 107228.

[4] G. Chryssolouris, D. Mavrikios, N. Papakostas, D. Mourtzis, G. Michalos, K. Georgoulias, Digital manufacturing: history, perspectives, and outlook, Proc. Inst. Mech. Eng. B 223 (5) (2009) 451–462.

[5] L. Lin, P.S. Kollipara, Y. Zheng, Digital manufacturing of advanced materials: Challenges and perspective, Mater. Today 28 (2019) 49–62.

[6] D.P. Möller, D.P. Möller, Digital manufacturing/industry 4.0, Guide Comput. Fundam. Cyber-Phys. Syst. Concepts Design Methods Appl. (2016) 307–375.

[7] Z. Vangelatos, H.M. Sheikh, P.S. Marcus, C.P. Grigoropoulos, V.Z. Lopez, G. Flamourakis, M. Farsari, Strength through defects: A novel Bayesian approach for the optimization of architected materials, Sci. Adv. 7 (41) (2021) eabk2218.

[8] L.R. Meza, S. Das, J.R. Greer, Strong, lightweight, and recoverable three-dimensional ceramic nanolattices, Science 345 (6202) (2014) 1322–1326.

[9] T. Fischer, C.M. Herr, Teaching generative design, in: Proceedings of the 4th Conference on Generative Art, Politechnico di Milano University Milan, 2001, pp. 147–160.

[10] F. Buonamici, M. Carfagni, R. Furferi, Y. Volpe, L. Governi, Generative design: an explorative study, Comput.-Aided Des. Appl. 18 (1) (2020) 144–155.

[11] J. Wu, O. Sigmund, J.P. Groen, Topology optimization of multi-scale structures: a review, Struct. Multidiscip. Optim. 63 (2021) 1455–1480.

[12] H.T. Kollmann, D.W. Abueidda, S. Koric, E. Guleryuz, N.A. Sobh, Deep learning for topology optimization of 2D metamaterials, Mater. Des. 196 (2020) 109098.

[13] J.N. Reddy, Introduction To the Finite Element Method, McGraw-Hill Education, 2019.

[14] R.D. Cook, et al., Concepts and Applications of Finite Element Analysis, John wiley & sons, 2007.

[15] G. Sinclair, J. Beisheim, A. Kardak, On the detection of stress singularities in finite element analysis, J. Appl. Mech. 86 (2) (2019) 021005.

[16] B. Zhu, The Finite Element Method: Fundamentals and Applications in Civil, Hydraulic, Mechanical and Aeronautical Engineering, John Wiley & Sons, 2018.

[17] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science 367 (6481) (2020) 1026–1030.

[18] P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, E. Hachem, A review on deep reinforcement learning for fluid mechanics, Comput. & Fluids 225 (2021) 104973.

[19] A. Barati Farimani, J. Gomes, V. Pande, Deep learning fluid mechanics, in: APS Division of Fluid Dynamics Meeting Abstracts, 2017, pp. E31–004.

[20] S. Ye, B. Li, Q. Li, H.-P. Zhao, X.-Q. Feng, Deep neural network method for predicting the mechanical properties of composites, Appl. Phys. Lett. 115 (16) (2019) 161901.

[21] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, M. Lei, Machine learning in materials science, InfoMat 1 (3) (2019) 338–358.

[22] F.E. Bock, R.C. Aydin, C.J. Cyron, N. Huber, S.R. Kalidindi, B. Klusemann, A review of the application of machine learning and data mining approaches in continuum materials mechanics, Front. Mater. 6 (2019) 110.

[23] H. Sasaki, H. Igarashi, Topology optimization accelerated by deep learning, IEEE Trans. Magn. 55 (6) (2019) 1–5.

[24] D.W. Abueidda, S. Koric, N.A. Sobh, Topology optimization of 2D structures with nonlinearities using deep learning, Comput. Struct. 237 (2020) 106283.

[25] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, rvtNat 521 (7553) (2015) 436–444.

[26] A. Javadi, T. Tan, M. Zhang, Neural network for constitutive modelling in finite element analysis, Comput. Assist. Mech. Eng. Sci. 10 (4) (2003) 523–530.

[27] X. Liu, F. Tao, H. Du, W. Yu, K. Xu, Learning nonlinear constitutive laws using neural network models based on indirectly measurable data, J. Appl. Mech. 87 (8) (2020) 081003.

[28] A. Oishi, G. Yagawa, Computational mechanics enhanced by deep learning, Comput. Methods Appl. Mech. Engrg. 327 (2017) 327–351.

[29] T. Spruegel, T. Schröppel, S. Wartzack, et al., Generic approach to plausibility checks for structural mechanics with deep learning, in: DS 87-1 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 1: Resource Sensitive Design, Design Research Applications and Case Studies, Vancouver, Canada, 21-25.08. 2017, 2017, pp. 299–308.

[30] A. Bhaduri, C.S. Meyer, J.W. Gillespie Jr., B.Z. Haque, M.D. Shields, L. Graham-Brady, Probabilistic modeling of discrete structural response with application to composite plate penetration models, J. Eng. Mech. 147 (11) (2021) 04021087.

[31] A. Bhaduri, D. Brandyberry, M.D. Shields, P. Geubelle, L. Graham-Brady, On the usefulness of gradient information in surrogate modeling: Application to uncertainty propagation in composite material models, Probab. Eng. Mech. 60 (2020) 103024.

[32] L. Lu, P. Jin, G.E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, 2019, arXiv preprint arXiv:1910.03193.

[33] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, Comput. Methods Appl. Mech. Engrg. 379 (2021) 113741.

[34] D.W. Abueidda, S. Koric, E. Guleryuz, N.A. Sobh, Enhanced physics-informed neural networks for hyperelasticity, Internat. J. Numer. Methods Engrg. (2022).

[35] A. Henkes, H. Wessels, R. Mahnken, Physics informed neural networks for continuum micromechanics, Comput. Methods Appl. Mech. Engrg. 393 (2022) 114790.

[36] J. Bai, T. Rabczuk, A. Gupta, L. Alzubaidi, Y. Gu, A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics, Comput. Mech. (2022) 1–20.

[37] E. Zhang, M. Yin, G.E. Karniadakis, Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging, 2020, arXiv preprint arXiv:2009.04525.

[38] S. Koric, A. Viswantah, D.W. Abueidda, N.A. Sobh, K. Khan, Deep learning operator network for plastic deformation with variable loads and material properties, Eng. Comput. (2023) 1–13.

[39] S. Korić, D.W. Abueidda, About applications of deep learning operator networks for design and optimization of advanced materials and processes, B & H Electr. Eng. 16 (s1) (2022) 1–6.

[40] J. He, S. Koric, S. Kushwaha, J. Park, D. Abueidda, I. Jasiuk, Novel DeepONet architecture to predict stresses in elastoplastic structures with variable complex geometries and loads, 2023, arXiv preprint arXiv:2306.03645.

[41] E. Zhang, M. Dao, G.E. Karniadakis, S. Suresh, Analyses of internal structures and defects in materials using physics-informed neural networks, Science advances 8 (7) (2022) eabk0644.

[42] D.W. Abueidda, Q. Lu, S. Koric, Meshless physics-informed deep learning method for three-dimensional solid mechanics, Internat. J. Numer. Methods Engrg. 122 (23) (2021) 7182–7201.

[43] H. Feng, P. Prabhakar, Difference-based deep learning framework for stress predictions in heterogeneous media, Compos. Struct. 269 (2021) 113957.

[44] V. Krokos, V. Bui Xuan, S.P. Bordas, P. Young, P. Kerfriden, A Bayesian multiscale CNN framework to predict local stress fields in structures with microscale features, Comput. Mech. 69 (3) (2022) 733–766.

[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[46] Z. Nie, H. Jiang, L.B. Kara, Stress field prediction in cantilevered structures using convolutional neural networks, J. Comput. Inf. Sci. Eng. 20 (1) (2020) 011002.

[47] B.P. Croom, M. Berkson, R.K. Mueller, M. Presley, S. Storck, Deep learning prediction of stress fields in additively manufactured metals with intricate defect networks, Mech. Mater. 165 (2022) 104191.

[48] J.R. Mianroodi, N.H. Siboni, D. Raabe, Teaching solid mechanics to artificial intelligence—a fast solver for heterogeneous materials, Npj Comput. Mater. 7 (1) (2021) 99.

[49] A. Bhaduri, A. Gupta, L. Graham-Brady, Stress field prediction in fiber-reinforced composite materials using a deep learning approach, Composites B 238 (2022) 109879.

[50] M.J. Buehler, FieldPerceiver: Domain agnostic transformer model to predict multiscale physical fields and nonlinear material properties through neural ologs, Mater. Today 57 (2022) 9–25.

[51] E.L. Buehler, M.J. Buehler, End-to-end prediction of multimaterial stress fields and fracture patterns using cycle-consistent adversarial and transformer neural networks, Biomed. Eng. Adv. 4 (2022) 100038.

[52] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014, arXiv:1406.2661.

[53] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: An overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.

[54] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, arXiv preprint arXiv:1411.1784.

[55] H. Jiang, Z. Nie, R. Yeo, A.B. Farimani, L.B. Kara, Stressgan: A generative deep learning model for two-dimensional stress distribution prediction, J. Appl. Mech. 88 (5) (2021).

[56] Z. Yang, C.-H. Yu, M.J. Buehler, Deep learning model to predict complex stress and strain fields in hierarchical composites, Sci. Adv. 7 (15) (2021) eabd7416.

[57] E. Hoq, O. Aljarrah, J. Li, J. Bi, A. Heryudono, W. Huang, Data-driven methods for stress field predictions in random heterogeneous materials, Eng. Appl. Artif. Intell. 123 (2023) 106267.

[58] L. Ning, Z. Cai, Y. Liu, W. Wang, Conditional generative adversarial network driven approach for direct prediction of thermal stress based on two-phase material SEM images, Ceram. Int. 47 (24) (2021) 34115–34126.

[59] M. Shahbazi, M. Danelljan, D.P. Paudel, L. Van Gool, Collapse by conditioning: Training class-conditional GANs with limited data, 2022, arXiv preprint arXiv:2201.06578.

[60] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning, PMLR, 2015, pp. 2256–2265.

[61] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.

[62] A.Q. Nichol, P. Dhariwal, Improved denoising diffusion probabilistic models, in: International Conference on Machine Learning, PMLR, 2021, pp. 8162–8171.

[63] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, Adv. Neural Inf. Process. Syst. 34 (2021) 8780–8794.

[64] Y. Song, C. Durkan, I. Murray, S. Ermon, Maximum likelihood training of score-based diffusion models, Adv. Neural Inf. Process. Syst. 34 (2021) 1415–1428.

[65] N. Liu, S. Li, Y. Du, A. Torralba, J.B. Tenenbaum, Compositional visual generation with composable diffusion models, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII, Springer, 2022, pp. 423–439.

[66] T. Salimans, J. Ho, Progressive distillation for fast sampling of diffusion models, 2022, arXiv preprint arXiv:2202.00512.

[67] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, J. Zhu, Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps, 2022, arXiv preprint arXiv:2206.00927.

[68] D. Shu, Z. Li, A.B. Farimani, A physics-informed diffusion model for high-fidelity flow field reconstruction, J. Comput. Phys. (2023) 111972.

[69] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, 2021, arXiv:2112.10752.

[70] M.J. Buehler, Modeling atomistic dynamic fracture mechanisms using a progressive transformer diffusion model, J. Appl. Mech. 89 (12) (2022) 121009.

[71] M.J. Buehler, Predicting mechanical fields near cracks using a progressive transformer diffusion model and exploration of generalization capacity, J. Mater. Res. 38 (5) (2023) 1317–1331.

[72] N.N. Vlassis, W. Sun, Denoising diffusion algorithm for inverse design of microstructures with fine-tuned nonlinear material properties, Comput. Methods Appl. Mech. Engrg. 413 (2023) 116126.

[73] E. Herron, X.Y. Lee, A. Balu, B.S.S. Pokuri, B. Ganapathysubramanian, S. Sarkar, A. Krishnamurthy, Generative design of material microstructures for organic solar cells using diffusion models, in: AI for Accelerated Materials Design NeurIPS 2022 Workshop, 2022.

[74] A.J. Lew, M.J. Buehler, Single-shot forward and inverse hierarchical architected materials design for nonlinear mechanical properties using an attention-diffusion model, Mater. Today 64 (2023) 10–20.

[75] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[77] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, 2020, arXiv preprint arXiv:2010.04159.

[78] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, H. Xia, End-to-end video instance segmentation with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8741–8750.

[79] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, D. Tao, A survey on vision transformer, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2023) 87–110, http://dx.doi.org/10.1109/TPAMI.2022.3152247.

[80] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[81] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, 2018, arXiv preprint arXiv:1803.02155.

[82] J. Gómez, N. Guarín-Zapata, Solidspy: 2dfinite element analysis with python, Parameters 50 (2018) 2.

[83] Y. Bao, H. Li, J. Ou, Emerging data technology in structural health monitoring: compressive sensing technology, J. Civ. Struct. Health Monit. 4 (2014) 77–90.

[84] J.L. Tabjula, S. Kanakambaran, S. Kalyani, P. Rajagopal, B. Srinivasan, Outlier analysis for defect detection using sparse sampling in guided wave structural health monitoring, Struct. Control Health Monit. 28 (3) (2021) e2690.

[85] F.-G. Yuan, S.A. Zargar, Q. Chen, S. Wang, Machine learning for structural health monitoring: challenges and opportunities, Sens. Smart Struct. Technol. Civ. Mech. Aerosp. Syst. 2020 11379 (2020) 1137903.

[86] A. Logg, G.N. Wells, DOLFIN: Automated finite element computing, ACM Trans. Math. Softw. 37 (2) (2010) 1–28.

[87] A. Logg, K.-A. Mardal, G. Wells, Automated Solution of Differential Equations By the Finite Element Method: The FEniCS Book, vol. 84, Springer Science & Business Media, 2012.

[88] M.W. Scroggs, J.S. Dokken, C.N. Richardson, G.N. Wells, Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes, ACM Trans. Math. Softw. 48 (2) (2022) 1–23.

[89] M.W. Scroggs, I.A. Baratta, C.N. Richardson, G.N. Wells, Basix: a runtime finite element basis evaluation library, J. Open Source Softw. 7 (73) (2022) 3982.

[90] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[91] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[92] D. Watson, J. Ho, M. Norouzi, W. Chan, Learning to efficiently sample from diffusion probabilistic models, 2021, arXiv preprint arXiv:2106.03802.

[93] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, 2020, arXiv preprint arXiv:2010.02502.

[94] J. Schmidt, P. Pelech, M. Dostalík, J. Malí k, 2D nonlinear elasticity, 2023, URL https://fenics-solid-tutorial.readthedocs.io/en/latest/2DNonlinearElasticity/2DNonlinearElasticity.html. (Accessed 19 July 2023).

[95] L. Weng, Attention? Attention!, 2018, lilianweng.github.io URL https://lilianweng.github.io/posts/2018-06-24-attention/.