Out of Domain Stress Prediction on a Dataset of Simulated 3D Polycrystalline Microstructures: Supplementary Material

Thomas Lu
Carnegie Mellon University
Pittsburgh, PA 15213
ttlu@cs.cmu.edu

Aarti Singh
Carnegie Mellon University
Pittsburgh, PA 15213
aarti@cs.cmu.edu

1 Description of Key Parameters of Experimentation

Model Architectures: We report the effectiveness of numerous architectures for our given task on both formulations. These models include a simple Convolutional Neural Network to predict single targets and fields, Visual Transformers, and U-Nets in conjunction to the additional deep methods listed below.

Transfer Learning: Transfer learning is a broad category of techniques that involve reusing knowledge from a source domain to improve performance on a target domain. In material science, this can involve fine-tuning pre-trained models (e.g., deep neural networks) on simulated data, or transferring knowledge from other related datasets. In these experiments, we experiment with transfer learning in multiple models. We experiment with using pretrained image classification networks such as VGG or ViT as an intermediate featurizer, which has shown positive results in past works involving similar materials.

Instance Reweighting: Instance reweighting methods assign different weights to samples from the source and target domains to mitigate the effects of domain shift. Techniques like importance weighting and domain-balanced loss functions can be used to emphasize the importance of certain samples during training. In some methods, where some small amount of data for the target texture is available for training, these samples – or augmented data based on these samples – were upweighted. When training labels for the target domain were not available, a separate classifier was trained to distinguish between textures, then points were weighted according to logits corresponding to the target texture. These classifiers were searched over the same architectures as our mean prediction models, but trained on classifying source texture. When training labels were available for a few samples of the target texture, these samples were simply upweighted.

Reconstruction Loss: In the training process for this approach, the model must reconstruct the input from an intermediate representation of the neural network in addition to predicting the output. This preserves relevant information in this representation, and can often act as a source of regularization.

Deep Domain Adaptation Methods: Domain Adversarial Neural Networks (DANNs) **?** and Multisource Domain Adaptation Networks (MDANs) **?** are neural networks that consist of three main components: a feature extractor, a label predictor, and a domain classifier. The domain classifier attempts to distinguish between source and target domain data using the extracted features, while the feature extractor and label predictor work together to minimize prediction error and maximize the error of the domain classifier. This adversarial training process encourages the feature extractor to learn domain-invariant representations, reducing the impact of domain shift. We also use an implementation of Deep CORAL as an additional training loss, which aligns the covariances of the features from each domain.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

2 Using a Single Microstructure from Target Texture

Another set of experiments included a single microstructure belonging to a target texture in the training data. This is in line with cases where only a single experimental or simulated sample needs to be obtained for use in a deep learning model. Models trained solely on this single microstructure are prone to overfitting, thus we experiment with a number of data augmentation techniques and model architectures to leverage the modestly-sized dataset of additional textures. Our experiments for this are shown in table 1. These experiments demonstrate that additional training on other textures is beneficial for learning stress relationships. A single microstructure likely does not have enough varied data for a machine learning model to learn generalizeable relationships, particularly due to the high levels of correlation between adjacent slices. Similar to our other stress field prediction formulation, CNN performs better out of domain than U-Net architectures.

3 Hyperparameters

All hyperparameters were chosen via experimentation over a wide range of complexities and choices. The reported parameters correspond to the best performing models.

3.1 Mean Prediction Models

CNN : 4 convolutional layers with kernel size 4, stride of 2. Channels: 3, 32, 64, 128, 256, 512. Batch norm after each convolution, followed by a ReLU transform. Reduced to hidden sizes of 768 to 2000 to the output, with a dropout of 0.2.

VGG : Taking features from any layers led to around the same performance. Final global representation fed into a network of hidden size 2000, then to the output, with a dropout of 0.2.

ViT : Trained using adapters on top of the standard pretrained model. Images were resized to 224 x 224.

3.2 Field Prediction Models

CNN : 4 Convolutional layers with kernel of 3, stride of 1, padding of 1. Channels: 3, 64, 128, 256, 512. The output of the final convolution is fed through a feed forward network with a hidden size of 300 to the output for each pixel, with a dropout of 0.2.

U-Net : Convolutional layers with kernel of 3, stride of 1, padding of 1. Channels: 3, 64, 128, 256, 512. The output of the final convolution is fed through a feed forward network with a hidden size of 300 to the output for each pixel, with a dropout of 0.2. Each layer is downsampled using max pooling to half the previous size.

ViT : Trained using adapters on top of the standard pretrained model. Images were resized to 224 x 224. The patch-wise features are expanded to 128x128 using transpose convolutional layers. The output of the final convolution is fed through a feed forward network with a hidden size of 300 to the output for each pixel, with a dropout of 0.2.

MDAN : The adversary is a feed forward network with a hidden size of 300 to the output for each pixel, with a dropout of 0.2. The weight for domain confusion loss was 1.0.

4 Additional Figures

Table 1: **Mean squared error predicting stress field across 11 direction using a single microstructure from the target texture**

Field Pred Model	Target Texture
Mean	148.95
U-Net with Source Data Only	147.81
U-Net with Source Data and Target Structure	125.18
U-Net with Source Data and Target Structure + Reflections	125.52
U-Net with Source Data and Target Structure + Reweighting	125.58
CNN with Target Structure Only	115.53
CNN with Source Data and Target Structure + Reweighting	97.47
CNN with MDAN	116.72

Figure 1: **Distribution of stress field by texture**. Note that only a few textures are shown for data clarity.

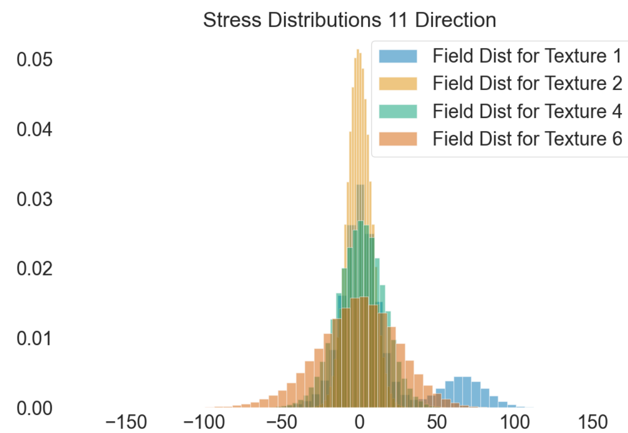

Figure 2: **Mean squared error plot for stress field prediction**

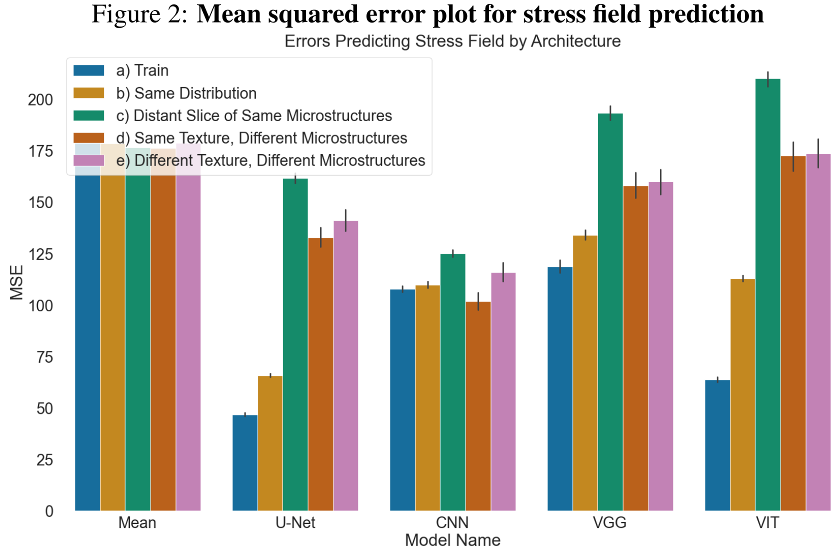

Figure 3: **Mean squared error over ablation between U-Net and CNN**

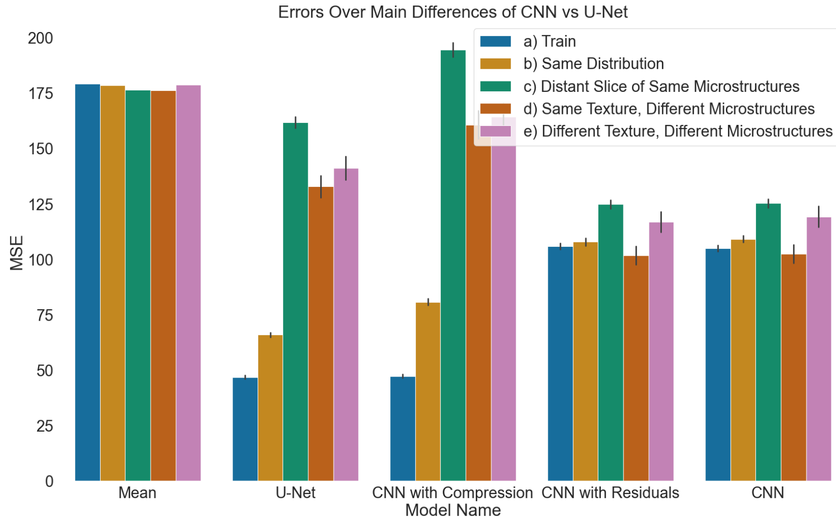

Figure 4: **Mean squared error over ViT patch sizes**. Only the model labeled as pretrained used pretrained weights, while others were trained on the dataset.

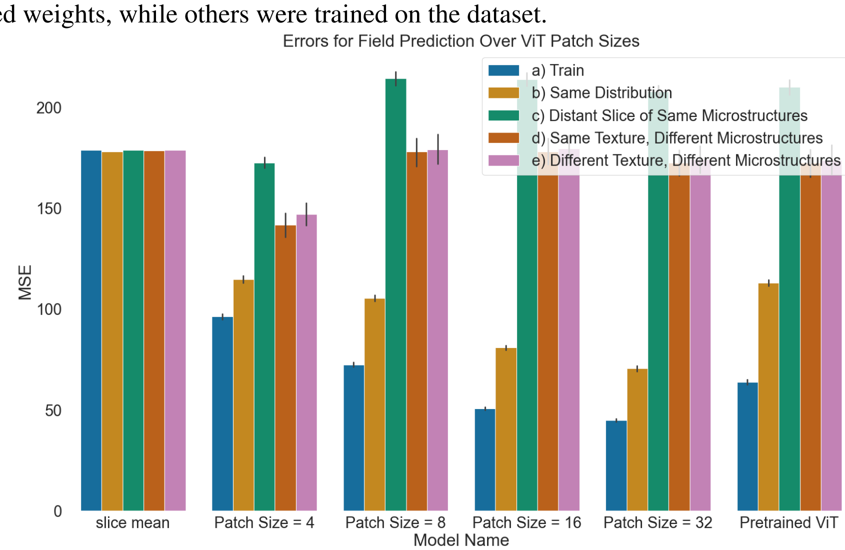

Figure 5: **Left: Histograms for the predicted values of a validation point from adjacent points of the training set compared with the true distribution. Right: Histograms for a distant slice of the training set compared with the true distribution.** The U-Net predicted with relatively low spread around a mean of around 0.75, while the CNN predicted with higher spread with a mean of 0.32. The true distribution had a mean centered at -0.27.