

Q-RBSA: High-Resolution 3D EBSD Map Generation Using An Efficient Quaternion Transformer Network

Devendra K. Jangid^{1, Cor}, Neal R. Brodnik², McLean P. Echlin³, Tresa M. Pollock³, Samantha H. Daly², and B.S. Manjunath¹

¹Electrical and Computer Engineering

²Mechanical Engineering

³Materials Department

^{Cor}dkjangid@ucsb.edu

Abstract

Gathering 3D material microstructural information is time-consuming, expensive, and energy-intensive. Acquisition of 3D data has been accelerated by developments in serial sectioning instrument capabilities; however, for crystallographic information, the electron backscatter diffraction (EBSD) imaging modality remains rate limiting. We propose a physics-based efficient deep learning framework to reduce the time and cost of collecting 3D EBSD maps. Our framework uses a quaternion residual block self-attention network (QRBSA) to generate high-resolution 3D EBSD maps from sparsely sectioned EBSD maps. In QRBSA, quaternion-valued convolution effectively learns local relations in orientation space, while self-attention in the quaternion domain captures long-range correlations. We apply our framework to 3D data collected from commercially relevant titanium alloys, showing both qualitatively and quantitatively that our method can predict missing samples (EBSD information between sparsely sectioned mapping points) as compared to high-resolution ground truth 3D EBSD maps.

1 Introduction

In the pursuit of the development of materials for extreme environments, 3D microstructural information is essential input for structure-property models [1]. Many engineering materials are polycrystalline, meaning they are composed of many smaller crystals called grains, and the arrangement of these grains impacts their thermomechanical properties. To collect crystallographic microstructure information, 3D microscopy techniques have been developed that span lengthscales from nanoscale to mesoscale [2]. These experiments require costly or challenging to access equipment, like synchrotron light sources for high X-ray fluxes [3, 4, 5], precise automated robotic mechanical polishing and imaging [6, 7], or high-energy ion beams and/or short pulse lasers coupled to electron microscopes [8, 9]. Recent advancements in 3D experimentation have greatly reduced the time required for other aspects of data collection, but serial sectioning methods (where material is progressively removed from the sample between images) predominately rely on electron backscatter diffraction (EBSD) imaging for collecting crystallographic information, which is three orders of magnitude slower per pixel (ms per pixel) than standard secondary electron or backscatter electron imaging (μ s per pixel).

EBSD is a scanning electron microscope (SEM) imaging modality that maps crystal lattice orientation by analyzing Kikuchi diffraction patterns that are formed when a focused electron beam is scattered by the atomic crystal structure of a material according to Bragg’s law. A grid of Kikuchi patterns is collected by scanning the electron beam across the sample surface. These patterns are then indexed to form a grid of orientations, which are commonly represented as images in RGB color space using inverse pole figure (IPF) projections. EBSD maps are used to determine the microstructural properties of materials such as texture, orientation gradients, phase distributions, and point-to-point orientation correlations, all of which are critical for understanding material performance [10]. However, scaling EBSD scans to 3D by serial sectioning is a time-consuming and energy-intensive process, often requiring hundreds of millions of EBSD patterns to be collected per sample - motivating methods to reduce the number of required points, such as smart or sparse sampling [11]

[12, 13], or machine learning super-resolution [14]. In these methods, missing information can be inferred using interpolation-based algorithms (bicubic, bilinear, or nearest neighbor) or data-based learning. Recent progress in computer vision [14, 15, 16] has shown that the generation of missing samples/data with data-based learning outperforms traditional interpolation algorithms for RGB images. However, unlike RGB images, EBSD maps have embedded crystallography, so existing learning-based methods are not well suited to generate missing EBSD data.

In our previous work [17], we developed a deep learning framework for 2D super-resolution that utilized an orientationally-aware, physics-based loss function to generate high-resolution (HR) EBSD maps from experimentally gathered low-resolution (LR) maps. This approach allowed for significant gains in 2D resolution, but expansion to 3D remained difficult due to data availability limitations (3D EBSD is expensive and time consuming to gather). To address this, here we have designed a 3D deep learning framework based on quaternion convolution neural networks with self-attention alongside physics-based loss to super-resolve high resolution 3D maps using as little data as possible. Using real-valued convolution for quaternion-based data has been shown to be inefficient and has loss in the inter-channel relationship [18]; leading to longer training times and larger data burdens. We demonstrate that a quaternion-valued neural network is more efficient and produces better results than real-valued convolution neural networks such as those used in previous work [17].

The crystallographic material information contained in EBSD maps is generally expressed in the form of crystal orientation spatially resolved at each pixel or voxel. These orientations, like other rotational data, can be expressed unambiguously using quaternions. They therefore can be incorporated into network architecture as prior information by using quaternion-valued convolution for local-level correlation, rather than real-valued or complex-valued convolution. The basic component in traditional CNN-based architectures is real-valued convolutional layers, which extract high-dimensional structural information using a set of convolution kernels. This approach is well-suited for unconstrained image data like RGB, but when convolution kernels fail to account for strict inter-channel dependen-

cies where present, the result is greater learning complexity. Some successful efforts have been made to design lower-complexity architectures by extending real-valued convolution to complex-valued convolution [19, 20] and quaternion-valued convolution [21, 22, 23] in the field of robotics [24], speech and text processing [18], computer graphics [25, 26], and computer vision [23, 27]. Although these convolution layers are useful to learn local correlations, they struggle to learn long-range correlations, whereas transformer-based architectures have recently shown significant success in learning long-range correlations in natural language [28] and high-level vision tasks [16, 29]. However, the computational complexity of transformer-based architectures grows quadratically with the spatial resolution of input images due to self-attention layers, so transformers alone are not well-suited for restoration tasks. However, recent work by Zamir [16] proposed self-attention across channel dimensions to reduce complexity from quadratic to linear with progressive learning for image restorations and showed superior results to convolution-based architecture alone.

Inspired by this idea, we propose the use of quaternion self-attention with progressive learning to incorporate long-range material relationships into super-resolved EBSD maps, while having linear computational scaling with the spatial size of the map. Progressive learning refers to having variable patch sizes instead of fixed patch sizes during training, which is relevant for most engineered material microstructures, where important features can span across length scales (and patch sizes). Titanium alloys, for example, are well-known to have many different microstructural variants accessible via processing, resulting in varying grain size and morphology. For the two alloys presented here, the Ti-6Al-4V variant has smaller equiaxed grains, while the Ti-7Al alloy has much larger grain size, so applying a fixed patch size would be sub-optimal across these two materials. To enforce long-range learning among these grain features, we used progressive patch sizes starting from 16 to 100 during the training of the network. Training behaves in a similar fashion to curriculum learning processes where the network starts with a simpler task and gradually moves to learning more complex ones.

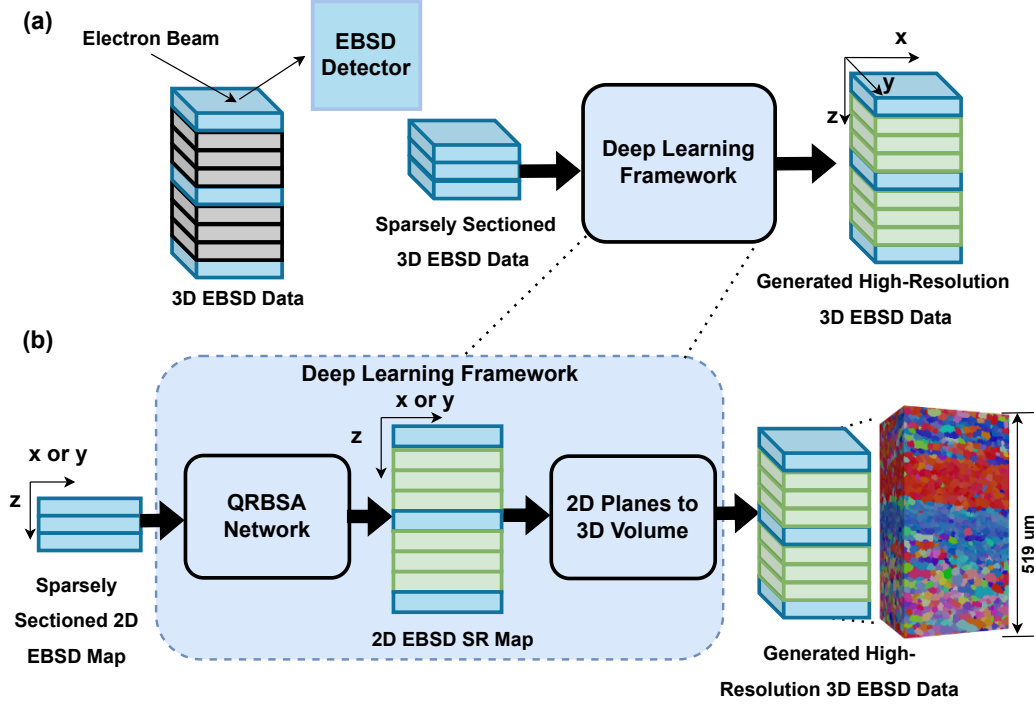


Figure 1: Q-RBSA EBSD RESOLUTION ENHANCEMENT FRAMEWORK: In the experimental pipeline shown in (a), material researchers collect EBSD orientation information for each (x,y) coordinate in a given sectioning plane, and then remove material using laser ablation or robotic polishing to reach the next plane in the z direction to build a 3D volume. In our framework, researchers collect EBSD information from a reduced set of points (blue planes), omitting some planes that would normally be gathered (gray planes). The missing information (green planes) are then generated in 2D as a series of (x,z) or (y,z) planes by our quaternion-based, physics-informed deep learning framework, shown in (b). Here, the network takes advantage of orthogonal independence to efficiently generate 3D volumes using less data, as large amounts of EBSD are costly and the choice of serial sectioning direction has minimal impact on the resultant final volume.

2 Results

2.1 Deep Learning Framework:

The objective of our framework is to generate missing sample planes from experimental 3D EBSD data that is sparse along the z-axis. In this approach, material researchers collect sparsely sectioned 3D EBSD data (blue planes) as shown in figure 1 (a), due to the high cost associated with serial sectioning and collecting 3D EBSD data at higher resolution. Ideally, a 3D deep learning framework would be designed to generate the missing planes (gray planes), but experimental EBSD data is costly to gather, so available 3D data is extremely limited. Additionally, 3D neural networks require more learned parameters, which, with limited available data, increases the likelihood of overfitting. Instead of a full 3D architecture, a deep learning network is implemented on 2D EBSD maps orthogonal to the sectioned planes, shown as the xz or yz planes in figure 1 (b). Our network takes sparsely sectioned xz or yz EBSD maps as input to generate the missing rows normal to the z-axis. The generated 2D EBSD maps are then combined into a 3D volume. EBSD collection is a point-based scanning method that is directionally independent; therefore missing z rows can be generated from xz or yz EBSD maps, and two 3D volumes can be formed from each sparsely sectioned dataset.

2.2 Network Architecture:

Although EBSD maps are visualized similarly to RGB images, they are multidimensional maps with inter-channel relationships, where crystal orientation is described using Euler angles, quaternions, matrices, or axis-angle pairs. Our previous work [17] demonstrated that quaternion representation is well-suited to orientation expression for loss function design, due to its efficient rotation simplification and avoidance of ambiguous representation. However, we previously used real-valued convolution layers to learn features, which is sub-optimal for EBSD orientation maps, as inter-channel quaternion relations are lost. Generally speaking, convolution networks provide local connectivity and translation equivariance, which are desirable properties for images, but if additional feature correlations are going to be learned efficiently, it is critical to encode relevant structural modalities into the network architec-

ture and loss function. Real-valued convolution can still learn quaternion inter-channel information, but it requires extra network complexity, and by consequence, additional data to inform that complexity. Here, the use of quaternion convolution efficiently encodes prior orientation information into kernels, and also has the advantage of reducing the number of trainable parameters by 4, as detailed in the supplement.

Quaternion convolution neural networks (QCNN) [30] use kernels which take the Hamilton product between the feature vectors rather than just computing correlations between features, as is done in real-valued convolution. In quaternion convolution, the feature output of each 2D convolution layer is split into four parts: the first part is the real component, and the remaining three parts are the complex vector component. Quaternion algebra is ensured by manipulating matrices of real numbers. In a QCNN, the convolution of a quaternion filter matrix with a quaternion vector is the Hamilton product. The Hamilton product between two quaternion vectors $p = (p_0, \vec{p})$ and $q = (q_0, \vec{q})$ is defined as:

$$(p_0, \vec{p}) \otimes (q_0, \vec{q}) = (p_0 q_0 - \vec{p} \cdot \vec{q}, p_0 \vec{q} + q_0 \vec{p} + \vec{p} \otimes \vec{q}) \quad (1)$$

Where, $\vec{p} = p_1 i + p_2 j + p_3 k$ and $\vec{q} = q_1 i + q_2 j + q_3 k$. Motivated by [16], we have designed a quaternion-based self-attention layer with a gated feed forward network in our network architecture, called QRBSA: Quaternion Residual Block Self-Attention as shown in figure 2. This efficient neural network architecture generates high-resolution EBSD maps from 2D input maps that are sparsely sectioned along the z-direction.

Introducing non-linearity through an activation function is not straightforward for quaternions, as the only functions that satisfy the Cauchy-Riemann-Fueter (CRF) equations in the quaternion domain are linear or constant [31]. However, locally analytic quaternion activation functions have been adapted for use in QNNs with standard backpropagation algorithms [32, 33]. There are two classes of these quaternion-valued activation functions: fully quaternion-valued functions and split functions. Fully quaternion-valued activation functions are an extension to the hypercomplex domain of real-valued functions, such as sigmoid or hyperbolic tangent functions. Despite their better performance [34], careful training is

needed due to the occurrence of singularities that can affect performance. To avoid this, split activation functions [35, 34] have been presented as a simpler solution for QNNs. In split activation functions, a conventional real-valued function is applied component-wise to a quaternion, alleviating singularities while holding true the universal approximation theorem as demonstrated in [35]. Through their simplicity, split activation functions map quaternion space back to real-valued space by ignoring the nature of the relationship that exists between the components. This is captured in the following relationship,

$$\alpha((p_0, \vec{p})) = f(p_0) + f(p_1)i + f(p_2)j + f(p_3)k \quad (2)$$

where α corresponds to a split function and $\vec{p} = p_1i + p_2j + p_3k$, and f represents any standard and real-valued activation function such as Sigmoid, Tanh, and ReLU. In our experiments, we used ReLU activation. Our network architecture shown in figure 2 consists of three parts: a shallow feature extraction module, a deep feature extraction module, and an upsampling and reconstruction module.

Shallow Feature Extractor: This module uses a single quaternion convolution layer to reduce the spatial size of sparsely sectioned EBSD maps, while extracting shallow features.

$$F_0 = H_{SF}(I_{LR}) \quad (3)$$

Here, I_{LR} is a sparsely sectioned 2D EBSD map and $H_{SF}(\cdot)$ is a single quaternion convolution layer of kernel size 3×3 , which has 4 input channels and 128 output channels. The generated shallow features (F_0) are given to the deep feature extractor module (H_{DF}).

Deep Feature Extractor: To learn from sparsely sectioned EBSD maps, our deep feature extractor module uses stacked quaternion residual self-attention (QRSA) blocks to extract high-frequency information and long skip connection to bypass low-frequency information. Residual blocks allow for a deeper network architecture, which provides a larger receptive field and better training stability. In our QRSA module, we use both CNN and transformer ideas to combine the effectiveness of the locality of CNNs with the expressivity of

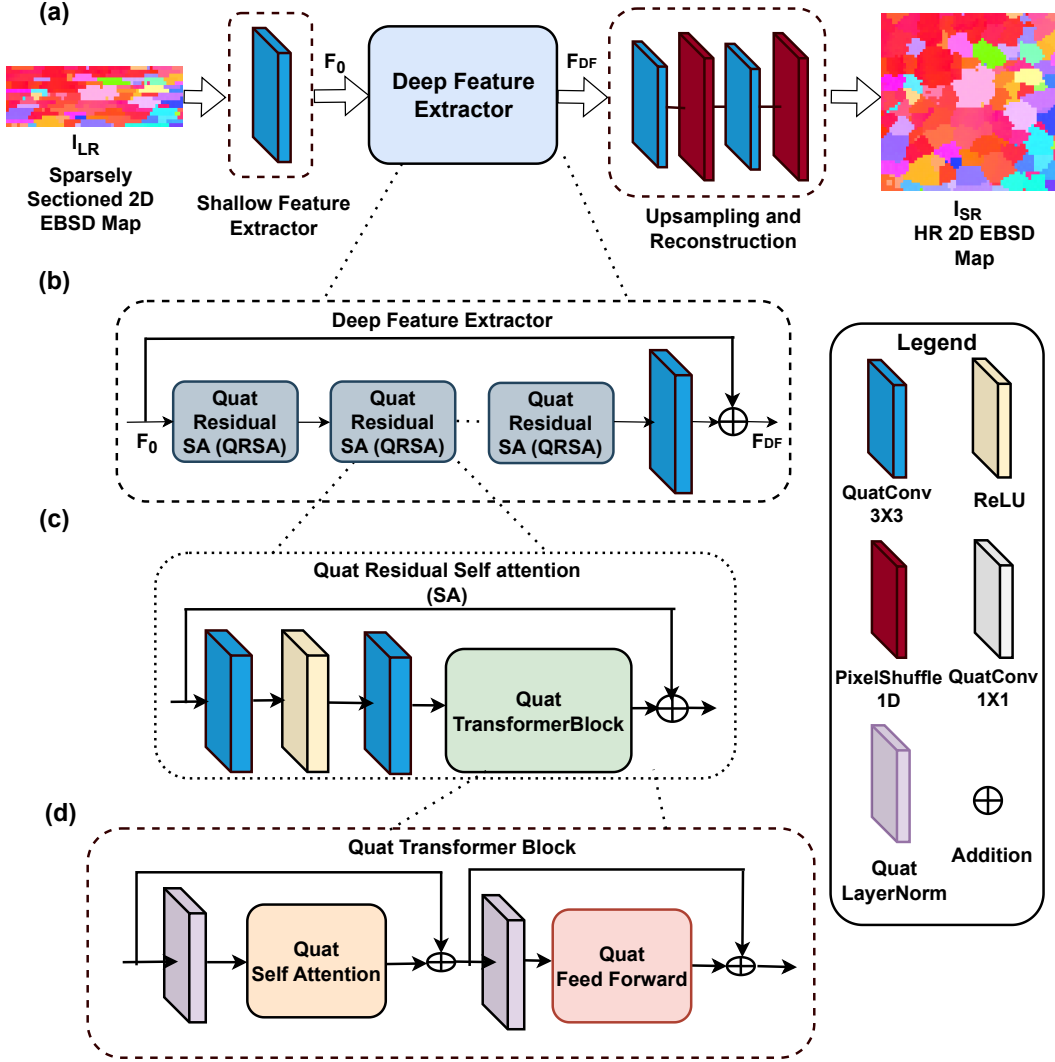


Figure 2: QUATERNION RESIDUAL BLOCK SELF-ATTENTION (QRBSA) NETWORK: A sparsely sectioned 2D EBSD map is given to the QRBSA network (a) to generate a high-resolution 2D EBSD map. QRBSA consists of three parts: a Shallow feature extractor, a Deep feature extractor, and Upsampling and Reconstruction. The deep feature extractor uses a residual architecture (b) where residual self-attention blocks (c) are modified with quaternion convolution layers and transformer blocks (d) to efficiently handle orientation data. Quaternion convolution is used to learn local-level relationships, while quaternion transformer blocks learn the global statistics of feature maps. Pixelshuffle layer, modified for 1-dimensional upsampling, is used in the upsampling and reconstruction block to upsample feature maps.

transformers that enables them to synthesize high-resolution EBSD maps. The CNN structure offers local connectivity and translation equivariance, allowing transformer components to freely learn complex and long-range relationships. Each quaternion residual self-attention (QRSA) block consists of two quaternion convolution layers and a piece-wise ReLU activation between them, and a quaternion transformer block. The quaternion convolution layers with piece-wise ReLU activation help in learning the local structure of extracted shallow features, while the quaternion transformer block captures long-range correlations among features. The short-skip connection is useful to bypass low-frequency information during training.

$$F_{DF} = H_{DF}(F_0) \quad (4)$$

$$H_{DF} = QConv \circ QRSA_n \circ QRSA_{n-1} \circ \dots QRSA_i \dots \circ QRSA_0 + I \quad (5)$$

where, $H_{DF}(\cdot)$ is a deep feature extractor module, and F_{DF} is a 128 channels feature map which goes to the upscale and reconstruction module. $QConv$ is a quaternion convolution, $QRSA_i$ is a i^{th} quaternion residual block, and I is a long skip connection.

Quaternion Transformer Block: The standard transformer architecture [28] consists of a self-attention layer, a feedforward network, and layer normalization. The original transformer architectures [28, 36] are not suitable for restoration tasks due to the requirement of quadratic complexity of spatial size $\mathcal{O}(W^2H^2)$, where W, H is the spatial size of images or EBSD maps. Similar to the approach of [16], we compute attention maps across the features dimension, which reduces the problem to linear complexity. However instead of depthwise convolution, we use quaternion convolution, which can be considered as a combination of depthwise convolution and group convolution, but with four-dimensional quaternion constraints. We have also incorporated an equivalent quaternion-based gating mechanism into the feedforward network within the transformer, and the traditional convolution used in [16] has been replaced with quaternion convolution layers to account for EBSD data modalities. Layer normalization plays a crucial role in the stability of training in transformer architectures. The quaternion layer-normalization is equivalent to the real-valued one, but adapted

to quaternion algebra, and allows the building of deeper architectures by normalizing the output at each layer. From the normalized features, the quaternion self-attention layer first generates query (Q), key (K) and value (V) projections enriched with local context. After reshaping query and key projection to reshaped query (Q_r) and reshaped key (K_r), a transposed attention map (A) is generated. The refined feature map, which has global statistical information, is calculated from the dot product of the value projection (V_r) and the attention map (A).

$$\text{Attention (A)} = \text{Softmax}(K_r \cdot \frac{Q_r}{\alpha}) \quad (6)$$

$$\text{Quat-SelfAttention}(Q_r, K_r, V_r) = V_r \cdot \text{A} \quad (7)$$

$$\text{Gating}(X_2) = \text{GeLU}(W_1(X_2)) \odot W_2(X_2) \quad (8)$$

Where $\odot$ represents elementwise multiplication, and W_i is a combination of quaternion convolution layers with kernel size 1 and 3, respectively.

Upsampling and Reconstruction: The upsampling and reconstruction module has a 1D pixelshuffle layer and a quaternion convolution layer of kernel size 3. The original pixelshuffle layer [37] is designed for 2D upsampling, but we have modified it for 1D upsampling in our framework that generates information in z dimension. Each block of the upsampling and reconstruction module upsamples deep features by a factor of 2, with the number of blocks depending on the scaling factors.

$$F_{\uparrow} = H_{\uparrow}(F_{DF}) \quad (9)$$

$$I_{SR} = H_R(F_{\uparrow}) \quad (10)$$

Where, each block $H_{\uparrow}$ of the module has a 1D pixel-shuffle layer, and a quaternion convolution layer H_R of kernel size 3×3 .

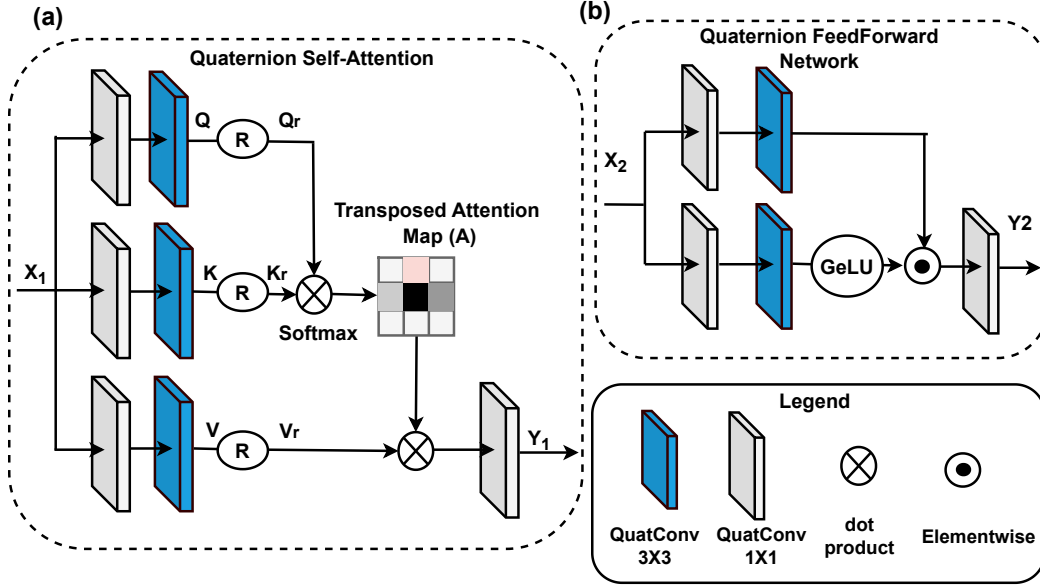


Figure 3: QUATERNION SELF ATTENTION: Self-Attention in (a) is computed using quaternion convolution across feature dimension instead of spatial dimension to reduce computational complexity to linear. An attention map (A) is calculated from reshaped query (Q_r) and reshaped key (K_r). A refined feature map is computed from the attention map (A) and reshaped value (V_r). QUATERNION FEED FORWARD NETWORK: Shown in (b), performs controlled feature transformation to allow useful information to propagate further using gated quaternion convolution.

2.3 2D to 3D:

The output of the QRBSA network is a 2D high-resolution EBSD map from a sparsely sectioned 2D EBSD map in the z direction. The 2D high-resolution EBSD maps are then combined to make a 3D volume. The missing z rows, as in figure 1 (b), can be generated either from xz plane (y_{normal}) or yz plane (x_{normal}). Therefore, there are two ways to form the 3D volume. In this work, we generated both 3D volumes separately, but we plan to design an algorithm in the future to combine the xz plane and yz plane information to make a single 3D volume.

2.4 Qualitative Output Comparison:

The sparsely sectioned 3D EBSD data is downsampled by scale factors of 2, 4 in the z dimension by removing the xy planes (z_{normal}) to reflect how EBSD resolution would be reduced in a serial sectioning experiment. Our network QRBSA is trained on 2D orthogonal planes (x_{normal} and y_{normal}) of paired sparsely sectioned EBSD maps and high-resolution EBSD maps, generating the high-resolution 2D maps in z dimension shown in figure 4. The most noticeable visual defects in 2D appear as pixel noise or short vertical lines, particularly around small grain features and high-aspect-ratio grains whose shortest axis is aligned with the z -direction. In addition to planar output analysis, we can also create 3D volumes from the sparsely sectioned xz planar (y_{normal}) or yz planar (x_{normal}) EBSD maps, and then sample the xy planes (z_{normal}) from these volumes to evaluate how well the QRBSA is inferring missing z -sample planes, as shown in figure 5. We can observe that our deep learning framework is able to completely predict omitted xy planes, comparably to the ground truth xy plane, with the exception of some shape variations around grain boundaries, particularly in Ti-6Al-4V. If the xy plane in figure 5 would have been omitted during experimental data collection, our framework would have estimated it with good accuracy.

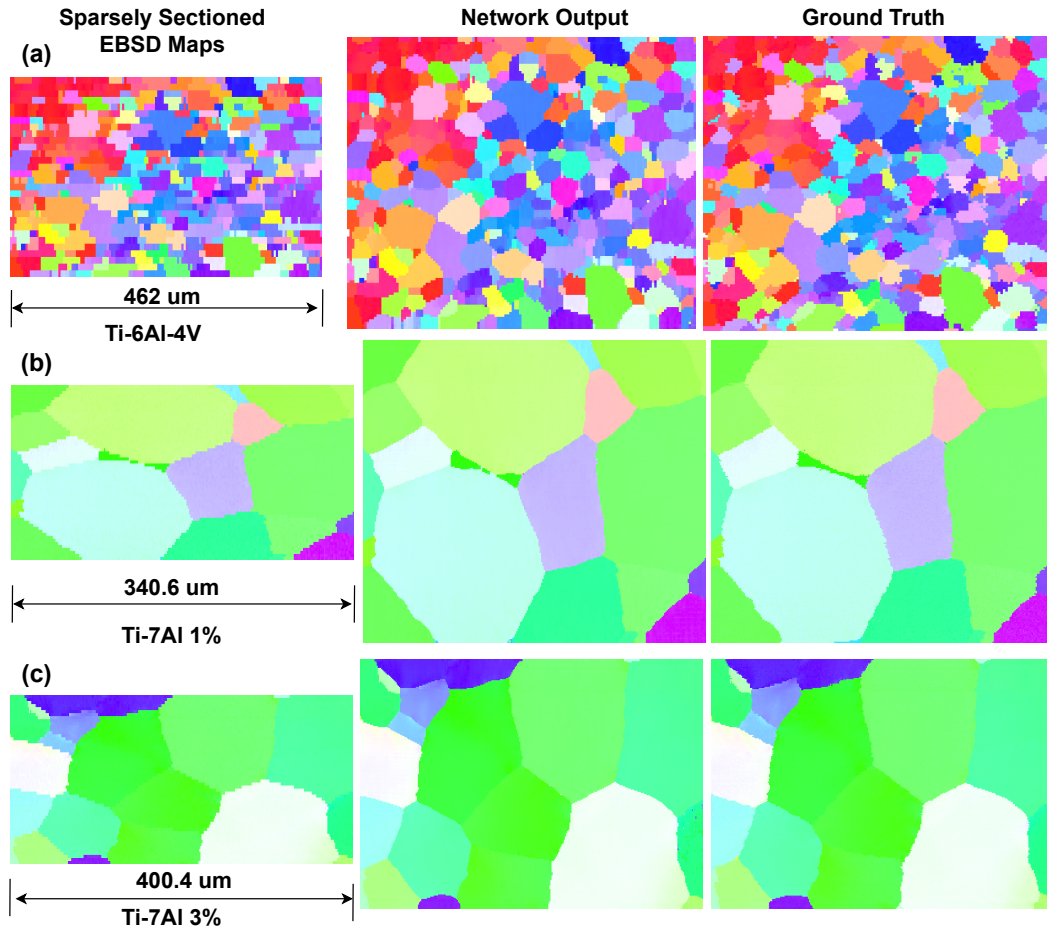


Figure 4: VISUAL COMPARISON OF NETWORK OUTPUT FOR EXAMPLE 2D EBSD MAPS WITH A SCALE FACTOR OF 4: The predicted EBSD maps (Network Output) from the QRBSA network are similar to the ground truth EBSD maps in for both the Ti-6Al-4V dataset (a) and both Ti-7Al datasets (b) and (c).

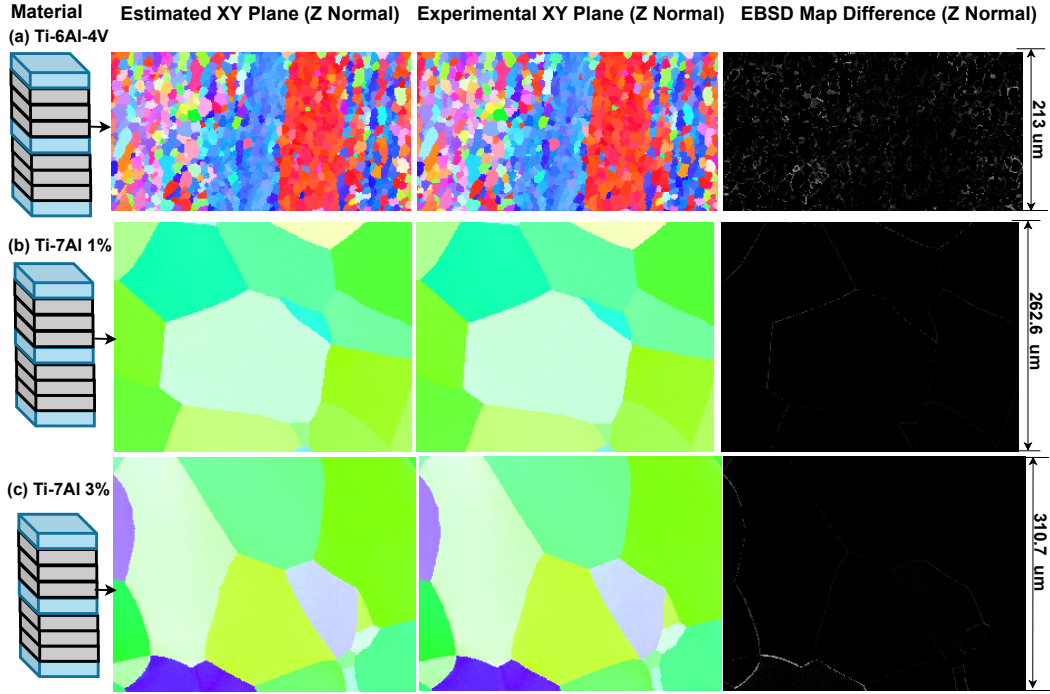


Figure 5: NEURAL NETWORK OUTPUT VS. GROUND TRUTH WITH DIFFERENCE MAP: The deep learning framework is able to estimate the missing xy planes due to sparse z-sampling (gray) with data that looks similar to the ground truth for Ti-6Al-4V in (a) and Ti-7Al in (b) and (c). The EBSD map difference column shows the intensity difference between ground truth and estimated EBSD maps, which indicate that learning grain shapes for Ti-6Al-4V is more difficult than for Ti-7Al, likely due to smaller grain size and more grain boundary regions.

Network	Trainable Parameters	Ti-6Al-4V		Ti-7Al 1%		Ti-7Al 3%	
		x2	x4	x2	x4	x2	x4
HAN	63,315,578	21.21/ 0.839	17.55/ 0.673	29.36/ 0.919	26.54/ 0.849	31.26/ 0.941	28.90/ 0.905
EDSR	6,355,460	21.49/ 0.86	18.09/ 0.718	30.06/ 0.944	27.34/ 0.907	32.16/ 0.958	29.57/ 0.932
QEDSR	1,593,092	21.44/ 0.861	18.04/ 0.710	29.89/ 0.935	27.22/ 0.904	32.05/ 0.95	29.52/ 0.93
QRBSA	5,952,782	21.60/ 0.870	18.2/ 0.730	30.20/ 0.946	27.48/ 0.908	32.36/ 0.96	29.65/ 0.94

Table 1: **PSNR/SSIM:** COMPARISON OF PSNR/SSIM AND COMPLEXITY FOR DIFFERENT NETWORKS FOR SCALE FACTORS 2 AND 4: Columns represent number of trainable parameters and PSNR/SSIM for different titanium datasets. A larger number is desired for both PSNR/SSIM.

2.5 Quantitative Output Comparison:

The image metrics of peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) were used to quantitatively evaluate the performance of the generated EBSD maps from the QRBSA network. We calculated the PSNR and SSIM of the Inverse Pole Figure (IPF) maps from the EBSD data for scale factors of 2 and 4 as shown in table 1. Higher PSNR and SSIM values represent more similarity with the ground truth. The PSNR/SSIM of Ti-6Al-4V is lower compared to Ti-7Al datasets due to its higher texture variability, wider range of orientations, and generally smaller grain features. We applied four different network architectures with different computational complexity to three different 3D EBSD datasets, as shown in table 1. Here, a simpler deep residual architecture (EDSR), outperforms a more complex holistic attention network (HAN) on EBSD data despite having significantly lower computational complexity. The amount of available EBSD data is significantly lower than open-source RGB image datasets, so simply increasing network complexity does not improve performance, as this added complexity demands additional training information and does not meaningfully consider relevant data modalities. QEDSR incorporates quaternion considerations in a similar architecture to EDSR, which greatly reduces in the number of network parameters, but also causes a slight drop in performance due to overall lack of complexity. We take advantage of this reduction in complexity to add in additional self-attention for better recognition of long-range patterns and global statistics. This

QRBSA network demonstrates the best performance on EBSD map restoration, while still maintaining lower complexity than state-of-the-art residual architectures for single-image super-resolution tasks.

3 Discussion

Both quantitative and qualitative results demonstrate that this physics-based deep learning framework can accurately estimate the missing xy planes (z_{normal}) of 3D EBSD data for multiple variants of titanium alloys, both with a coarser polycrystalline structure (Ti-7Al) and finer structure with stronger texture (Ti-6Al-4V). In 2D inferred EBSD planes show noise around small features, mostly in the form of point and line defects in the z -direction associated with grains whose overall shape information was lost due to omission of sample planes in low resolution. It is possible that a downsampling approach incorporating anti-aliasing could prevent this shape information loss [38], but this approach would not be reflective of actual experimental downsampling in 3D EBSD. This general shape loss effect, along with a larger number of small grains, varying local crystallographic texture, and a wide range of represented crystal orientations, made the Ti-6Al-4V the most difficult dataset for inference. This is further evidenced by a larger number of grain boundary differences for Ti-6Al-4V in figure 5, as well as a lower PSNR/SSIM score in table 1. Additional noise analysis for generated xy planes is shown in the supplement, and there is ongoing work to improve performance using 3D architectures and grain shape information [39] with adaptive multi-scale imaging in z dimension as more of this type of data becomes available.

When considering data beyond the use-case sets tested, the approach shown here is directly applicable to any serial sectioning technique for gathering 3D EBSD information, including FIB sectioning, laser ablation, and robotic serial sectioning [7, 6]. Further, data from other 3D grain mapping techniques that rely on synchrotron X-ray sources such as diffraction contrast tomography (DCT) [5] or high energy diffraction microscopy (HEDM) [4, 3] may also be applicable for the infrastructure presented here. Similar approaches to this may be particularly useful in lab source DCT experiments [40, 41], where the X-ray

source constraints limit grain mapping resolution in comparison to synchrotron sources. For example, one could use difficult to acquire synchrotron X-ray mapping experiments as HR data to train a network to inform LR X-ray mapping experiments collected more routinely at the laboratory.

In summary, we have designed a quaternion-convolution-based deep learning framework with crystallography physics-based loss to generate costly high-resolution 3D EBSD data from sparsely sectioned 3D EBSD data while accounting for the physical constraints of crystal orientation and symmetry. Alongside this, an efficient quaternion-based transformer block was developed to learn long-range trends and global statistics from EBSD maps. Using quaternion convolution instead of regular convolution is critical for crystallographic data, both in terms of output quality and neural network complexity, as reducing the number of trainable parameters enables transformer addition without major complexity burden (see table [1](#)). This framework can be directly applied to any experimental 3D EBSD approaches that rely on serial sectioning techniques to collect orientation information.

4 Methods

4.1 EBSD Datasets

EBSD maps represent crystal orientations collected at each physical pixel location in crystalline materials, which are fundamentally anisotropic and atomically periodic. Orientations for each pixel within the network learning environment are expressed in terms of quaternions of the form $q = q_0 + iq_1 + jq_2 + kq_3$. The quaternions are suitable to design a physics-based loss function for deep learning framework [\[17\]](#). To avoid redundancy in quaternion space (between q and $-q$), all orientations are expressed with their scalar q_0 as positive. For visualization according to established conventions, quaternions are reduced to the Rodrigues space fundamental zone based on space group symmetry, converted into Euler angles, and projected using inverse pole figure (IPF) projection using the open-source Dream3D software [\[42\]](#), as shown in our previous work [\[17\]](#). Ground truth 3D EBSD datasets were experimentally collected from titanium alloy samples: Ti-6Al-4V and Ti-7Al (one Ti-7Al sample

deformed in tension to 1%, and one to 3%), using a commercially-available rapid-serial-sectioning electron microscope referred to as the TriBeam [43, 8]. sparsely sectioned EBSD datasets are created by removing xy (z_{normal}) planes from the high-resolution ground truth with a downscale factor of 2 and 4 ($LR = \frac{1}{4}HR$ or $LR=\frac{1}{2}HR$). This is done to imitate the beam raster steps that would occur in a Tribeam experiment with more sparsely sectioned EBSD data, which would not influence the electron beam-material interaction volume at each location, but rather lead to greater raster spacing step sizes. More information about dataset pre-processing are given in the supplementary material.

4.2 Network Implementation and Output Evaluation

We use a learning rate of 0.0002, an Adam optimizer with $\beta_1 = 0.9$, $\beta_2=0.99$, ReLU activation, batch size of 4 and downscaling factor of 2 and 4. The patch size of HR EBSD maps is selected from [16, 32, 64, 100] during training of the network. The framework is implemented in PyTorch and trained on NVIDIA Tesla V100 GPU for 2000 epochs, which took approximately 100 hours. Once training is completed, inference time for a given 2D LR EBSD map is on the order of less than one second for an imaging area that would normally take about 10 minutes to gather manually.

4.3 Loss:

The QRBSA network is trained using a physics based loss function [17], which uses rotational distance approximation loss with enforced hexagonal crystal symmetry (HCP). Rotational distance loss computes the misorientation angles between the predicted and ground truth EBSD map in the same manner that they would be measured during crystallographic analysis, with approximations to avoid discontinuities at the edge of the fundamental zone. The rotational distance between two quaternions can be computed as the following:

$$\theta = 4 \sin^{-1} \left(\frac{d_{euclid}}{2} \right) \quad (11)$$

where, $d_{\text{euclid}} = \|q_1 - q_2\|_2$. While d_{euclid} is Lipschitz, the gradient of θ goes to ∞ as $d_{\text{euclid}} \rightarrow 2$. To address this issue during neural network training, a linear approximation was computed at $d_{\text{euclid}} = 1.9$, and utilized for points > 1.9 .

4.4 Progressive Learning:

In our previous work [17], we used a fixed patch size of dimension 64×64 for training the CNN based architectures which help in learning local correlations. However, self-attention is required to have larger patch sizes, which aids in learning global correlations. Inspired from the work of Zamir [16], we use progressive patch samples from sizes of [16, 32, 64, 100] in the training process to learn global statistics. We start from a smaller patch size in early epochs and increase to a larger patch sizes in the later epochs. The progressive learning acts like the curriculum learning process where a network starts with a simple tasks and gradually moves to learning a more complex one.

5 Data Availability

Pre-trained versions of the network modules produced in this paper will be made publicly available through the BisQue cyberinfrastructure at <https://bisque2.ece.ucsb.edu>. Material datasets will be available by request at the discretion of the authors.

6 Code Availability

Architecture code will be made publicly accessible through GitHub.

References

- [1] N. R. Council, *Integrated Computational Materials Engineering: A Transformational Discipline for Improved Competitiveness and National Security*. Washington, DC: The National Academies Press, 2008.

- [2] M. P. Echlin, T. L. Burnett, A. T. Polonsky, T. M. Pollock, and P. J. Withers, “Serial sectioning in the SEM for three dimensional materials science,” *Current Opinion in Solid State and Materials Science*, vol. 24, no. 2, p. 100817, Apr. 2020. [Online]. Available: <https://doi.org/10.1016/j.cossms.2020.100817>
- [3] M. P. Miller, D. C. Pagan, A. J. Beaudoin, K. E. Nygren, and D. J. Shadle, “Understanding micromechanical material behavior using synchrotron x-rays and in situ loading,” *Metallurgical and Materials Transactions A*, vol. 51, no. 9, pp. 4360–4376, Jun. 2020. [Online]. Available: <https://doi.org/10.1007/s11661-020-05888-w>
- [4] J. V. Bernier, R. M. Suter, A. D. Rollett, and J. D. Almer, “High-energy x-ray diffraction microscopy in materials science,” *Annual Review of Materials Research*, vol. 50, no. 1, pp. 395–436, Jul. 2020. [Online]. Available: <https://doi.org/10.1146/annurev-matsci-070616-124125>
- [5] P. Reischig and W. Ludwig, “Three-dimensional reconstruction of intragranular strain and orientation in polycrystals by near-field x-ray diffraction,” *Current Opinion in Solid State and Materials Science*, vol. 24, no. 5, p. 100851, 2020.
- [6] D. J. Rowenhorst, L. Nguyen, A. D. Murphy-Leonard, and R. W. Fonda, “Characterization of microstructure in additively manufactured 316l using automated serial sectioning,” *Current Opinion in Solid State and Materials Science*, vol. 24, no. 3, p. 100819, Jun. 2020. [Online]. Available: <https://doi.org/10.1016/j.cossms.2020.100819>
- [7] M. G. Chapman, M. N. Shah, S. P. Donegan, J. M. Scott, P. A. Shade, D. Menasche, and M. D. Uchic, “AFRL additive manufacturing modeling series: Challenge 4, 3d reconstruction of an IN625 high-energy diffraction microscopy sample using multi-modal serial sectioning,” *Integrating Materials and Manufacturing Innovation*, vol. 10, no. 2, pp. 129–141, Jun. 2021. [Online]. Available: <https://doi.org/10.1007/s40192-021-00212-9>
- [8] M. P. Echlin, A. T. Polonsky, J. Lamb, R. Geurts, S. J. Randolph, A. Botman,

- and T. M. Pollock, “Recent developments in femtosecond laser-enabled TriBeam systems,” *JOM*, vol. 73, no. 12, pp. 4258–4269, Nov. 2021. [Online]. Available: <https://doi.org/10.1007/s11837-021-04919-0>
- [9] A. Garner, J. Donoghue, R. Geurts, Y. A. Aboura, B. Winiarski, P. B. Prangnell, and T. L. Burnett, “Large-scale serial sectioning of environmentally assisted cracks in 7xxx al alloys using femtosecond laser-PFIB,” *Materials Characterization*, vol. 188, p. 111890, Jun. 2022. [Online]. Available: <https://doi.org/10.1016/j.matchar.2022.111890>
- [10] A. J. Schwartz, M. Kumar, B. L. Adams, and D. P. Field, Eds., *Electron Backscatter Diffraction in Materials Science*. Springer US, 2009. [Online]. Available: <https://doi.org/10.1007/978-0-387-88136-2>
- [11] G. M. D. P. Godaliyadda, D. H. Ye, M. D. Uchic, M. A. Groeber, G. T. Buzzard, and C. A. Bouman, “A framework for dynamic image sampling based on supervised learning,” *IEEE Transactions on Computational Imaging*, vol. 4, no. 1, pp. 1–16, Mar. 2018. [Online]. Available: <https://doi.org/10.1109/tci.2017.2777482>
- [12] Y. Zhang, G. M. D. Godaliyadda, N. Ferrier, E. B. Gulsoy, C. A. Bouman, and C. Phatak, “Reduced electron exposure for energy-dispersive spectroscopy using dynamic sampling,” *Ultramicroscopy*, vol. 184, pp. 90–97, Jan. 2018. [Online]. Available: <https://doi.org/10.1016/j.ultramic.2017.10.015>
- [13] V. S. Tong, A. J. Knowles, D. Dye, and T. B. Britton, “Rapid electron backscatter diffraction mapping: Painting by numbers,” *Materials Characterization*, vol. 147, pp. 271–279, Jan. 2019. [Online]. Available: <https://doi.org/10.1016/j.matchar.2018.11.014>
- [14] Z. Wang, J. Chen, and S. C. Hoi, “Deep learning for image super-resolution: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 3365–3387, 10 2021. [Online]. Available: <https://doi.org/10.1109/TPAMI.2020.2982166>
- [15] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, “Second-order attention network for single image super-resolution,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 065–11 074.

- [16] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, “Restormer: Efficient transformer for high-resolution image restoration,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5728–5739.
- [17] D. K. Jangid, N. R. Brodnik, M. G. Goebel, A. Khan, S. Majeti, M. P. Echlin, S. H. Daly, T. M. Pollock, and B. Manjunath, “Adaptable physics-based super-resolution for electron backscatter diffraction maps,” *npj Computational Materials*, vol. 8, no. 1, p. 255, 2022.
- [18] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linares, R. De Mori, and Y. Bengio, “Quaternion convolutional neural networks for end-to-end automatic speech recognition,” *arXiv preprint arXiv:1806.07789*, 2018.
- [19] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. Pal, “Deep complex networks. arxiv preprint arxiv: 170509792,” 2017.
- [20] I. Aizenberg and A. Gonzalez, “Image recognition using mlmvn and frequency domain features,” in *2018 International joint conference on neural networks (IJCNN)*. IEEE, 2018, pp. 1–8.
- [21] N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura, “Quaternion neural network with geometrical operators,” *Journal of Intelligent & Fuzzy Systems*, vol. 15, no. 3-4, pp. 149–164, 2004.
- [22] H. Kusamichi, T. Isokawa, N. Matsui, Y. Ogawa, and K. Maeda, “A new scheme for color night vision by quaternion neural network,” in *Proceedings of the 2nd international conference on autonomous robots and agents*, vol. 1315, 2004.
- [23] T. Isokawa, N. Matsui, and H. Nishimura, “Quaternionic neural networks: Fundamental properties and applications,” in *Complex-valued neural networks: utilizing high-dimensional parameters*. IGI global, 2009, pp. 411–439.

- [24] X. Yun and E. R. Bachmann, “Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking,” *IEEE transactions on Robotics*, vol. 22, no. 6, pp. 1216–1227, 2006.
- [25] K. Shoemake, “Animating rotation with quaternion curves,” in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985, pp. 245–254.
- [26] D. Pletinckx, “Quaternion calculus as a basic tool in computer graphics,” *The Visual Computer*, vol. 5, no. 1, pp. 2–13, 1989.
- [27] X. Zhu, Y. Xu, H. Xu, and C. Chen, “Quaternion convolutional neural networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [29] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “Swinir: Image restoration using swin transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1833–1844.
- [30] C. J. Gaudet and A. S. Maida, “Deep quaternion networks,” in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.
- [31] T. Parcollet, M. Morchid, and G. Linarès, “A survey of quaternion neural networks,” *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2957–2982, 2020.
- [32] S. De Leo and P. Rotelli, “Local hypercomplex analyticity,” *arXiv preprint function/9703002*, 1997.
- [33] T. Isokawa, H. Nishimura, and N. Matsui, “Quaternionic multilayer perceptron with local analyticity,” *Information*, vol. 3, no. 4, pp. 756–770, 2012.
- [34] B. C. Ujang, C. Jahanchahi, C. C. Took, and D. Mandic, “Quaternion valued neural networks and nonlinear adaptive filters,” *IEEE Trans. Neural Netw*, 2010.

- [35] P. Arena, L. Fortuna, L. Occhipinti, and M. G. Xibilia, “Neural networks for quaternion-valued function approximation,” in *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS’94*, vol. 6. IEEE, 1994, pp. 307–310.
- [36] 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,” *arXiv preprint arXiv:2010.11929*, 2020.
- [37] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.
- [38] J. Jung, J. Na, H. K. Park, J. M. Park, G. Kim, S. Lee, and H. S. Kim, “Super-resolving material microstructure image via deep learning for microstructure characterization and mechanical behavior analysis,” *npj Computational Materials*, vol. 7, no. 1, Jun. 2021. [Online]. Available: <https://doi.org/10.1038/s41524-021-00568-8>
- [39] D. K. Jangid, N. R. Brodnik, A. Khan, M. G. Goebel, M. P. Echlin, T. M. Pollock, S. H. Daly, and B. Manjunath, “3d grain shape generation in polycrystals using generative adversarial networks,” *Integrating Materials and Manufacturing Innovation*, vol. 11, no. 1, pp. 71–84, 2022.
- [40] J. Oddershede, J. Sun, N. Gueninchault, F. Bachmann, H. Bale, C. Holzner, and E. Lauridsen, “Non-destructive characterization of polycrystalline materials in 3d by laboratory diffraction contrast tomography,” *Integrating Materials and Manufacturing Innovation*, vol. 8, no. 2, pp. 217–225, May 2019. [Online]. Available: <https://doi.org/10.1007/s40192-019-00135-6>
- [41] F. Bachmann, H. Bale, N. Gueninchault, C. Holzner, and E. M. Lauridsen, “3d grain reconstruction from laboratory diffraction contrast tomography,” *Journal of*

- Applied Crystallography*, vol. 52, no. 3, pp. 643–651, May 2019. [Online]. Available: <https://doi.org/10.1107/s1600576719005442>
- [42] M. A. Groeber and M. A. Jackson, “DREAM.3D: A digital representation environment for the analysis of microstructure in 3D,” *Integrating Materials and Manufacturing Innovation*, vol. 3, no. 1, pp. 56–72, Apr. 2014. [Online]. Available: <https://doi.org/10.1186/2193-9772-3-5>
- [43] M. P. Echlin, M. Straw, S. Randolph, J. Filevich, and T. M. Pollock, “The TriBeam system: Femtosecond laser ablation in situ SEM,” *Materials Characterization*, vol. 100, pp. 1–12, Feb. 2015. [Online]. Available: <https://doi.org/10.1016/j.matchar.2014.10.023>
- [44] M. P. Echlin, T. L. Burnett, A. T. Polonsky, T. M. Pollock, and P. J. Withers, “Serial sectioning in the SEM for three dimensional materials science,” *Current Opinion in Solid State and Materials Science*, vol. 24, no. 2, p. 100817, Apr. 2020. [Online]. Available: <https://doi.org/10.1016/j.cossms.2020.100817>
- [45] S. Hémerly, A. Naït-Ali, M. Guéguen, J. Wendorf, A. Polonsky, M. Echlin, J. Stinville, T. Pollock, and P. Villechaise, “A 3D analysis of the onset of slip activity in relation to the degree of micro-texture in Ti-6Al-4V,” *Acta Materialia*, vol. 181, pp. 36–48, Dec. 2019. [Online]. Available: <https://doi.org/10.1016/j.actamat.2019.09.028>
- [46] M. Charpagne, J. C. Stinville, A. T. Polonsky, M. P. Echlin, and T. M. Pollock, “A multi-modal data merging framework for correlative investigation of strain localization in three dimensions,” *JOM*, vol. 73, no. 11, pp. 3263–3271, Sep. 2021. [Online]. Available: <https://doi.org/10.1007/s11837-021-04894-6>
- [47] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

7 Acknowledgements

This research is supported in part by NSF award number 1664172. N.Brodnik gratefully acknowledges financial support from NSWC Grant (N00174-22-1-0020). The authors gratefully acknowledge Patrick Callahan, Toby Francis, Andrew Polonsky, and Joseph Wendorf for collection of the 3D Ti-6Al-4V and Ti-7Al datasets. The MRL Shared Experimental Facilities are supported by the MRSEC Program of the NSF under Award No. DMR 1720256; a member of the NSF-funded Materials Research Facilities Network (www.mrfln.org). Use was also made of computational facilities purchased with funds from the National Science Foundation (CNS-1725797) and administered by the Center for Scientific Computing (CSC). The CSC is supported by the California NanoSystems Institute and the Materials Research Science and Engineering Center (MRSEC; NSF DMR 1720256) at UC Santa Barbara. Use was made of the computational facilities purchased with funds from the National Science Foundation CC* Compute grant (OAC-1925717) and administered by the Center for Scientific Computing (CSC). The ONR Grant N00014-19-2129 is also acknowledged for the titanium datasets. The authors of this work declare no competing financial or non-financial interests.

8 Author Contributions

- D.K. Jangid: Development of complete framework, Design and coding of complete network architecture, Preprocessing of datasets for training the network, All experiments to make our idea work, Qualitative and quantitative evaluation (image metrics) of results, Manuscript Preparation
- N.R. Brodnik: Initial preparation of training, validation, and test datasets , Feedback on generated results, Mentoring, Conception of Ideas, Manuscript Preparation
- M.P. Echlin: Preparation of datasets, Conception of ideas, Mentoring, Feedback on generated results from network architecture, Manuscript Preparation
- S.H. Daly (Co-PI): Conception of ideas, Mentoring, Funding Acquisition, Manuscript

Preparation

- T.M. Pollock (Co-PI): Conception of ideas, Mentoring, Funding Acquisition, Manuscript Preparation
- B.S. Manjunath (Co-PI): Conception of ideas, Mentoring, Funding Acquisition, Manuscript Preparation
- All authors contributed to the writing of this manuscript with writing efforts being led by D.K. Jangid, N.R. Brodnik and M.P. Echlin.

9 Competing Interests

The Authors declare that there are no competing interests.

10 Supplementary Material

10.1 Quaternion Convolution Neural Network:

A quaternion q is a four component number of the form $q = q_0 + iq_1 + jq_2 + kq_3$, where basis vectors (i, j, k) satisfy the following relationship:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (12)$$

The quaternion can be represented as matrix of real numbers.

$$\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \quad (13)$$

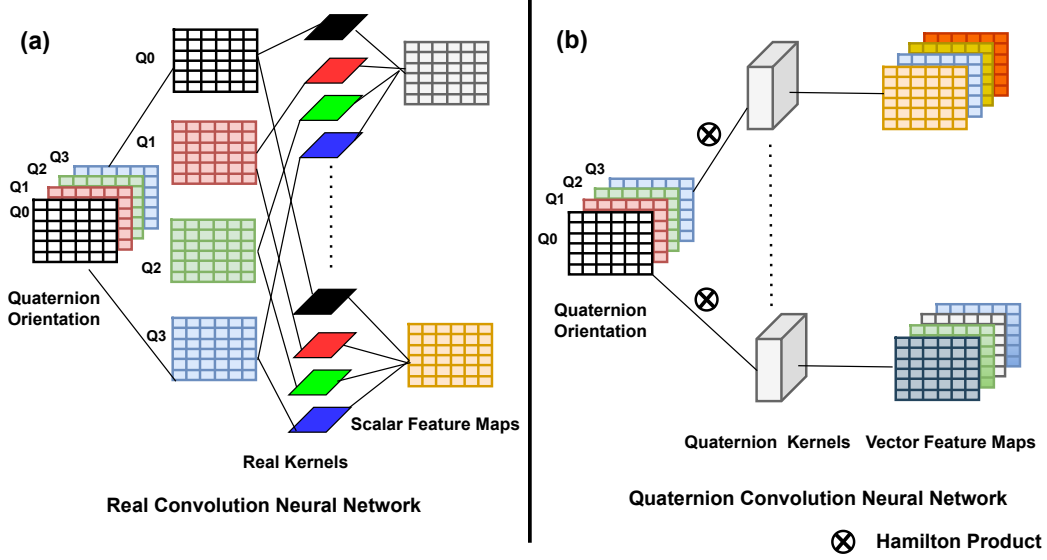


Figure 6: QUATERNION CONVOLUTION: In a real convolution neural network, correlation, a scalar quantity, is computed between independent kernels and channel dimension of quaternion feature maps. In a quaternion convolution neural network, the Hamilton product, a vector quantity, is computed between quaternion kernels and quaternion feature input maps. The parameters in quaternion kernel is restricted according to quaternion algebra [31].

In a Quaternion Convolution Neural Network (QCNN), the convolution of a quaternion filter matrix with a quaternion vector is Hamilton product. The Hamilton product can also be represented as a matrix of real numbers.

$$p \otimes q = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} * \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (14)$$

The difference between real convolution neural network (RCNN) and quaternion convolution neural network (QCNN) lies on how a basic convolution operation is performed as shown in figure 6. In RCNN, scalar feature maps are computed by computing correlations between channels of feature maps and kernel filters independently without considering

inter-channel relationship. But, vector feature maps are generated using the Hamilton product between feature maps and quaternion kernels in QCNN. Both quaternion kernels and Hamilton products take inter-channel relationships into consideration.

10.2 Data Preprocessing

High-Resolution (HR) Ground Truth: The ground truth data is experimental 3D EBSD data gathered from the titanium alloys, Ti-6Al-4V and Ti-7Al (one Ti-7Al sample deformed in tension to 1% and one to 3%), using a commercially-available rapid-serial-sectioning electron microscope known as the Tribeam [43, 44]. The Ti-6Al-4V dataset, shown in figure 7(a), is of pixel size $346 \times 142 \times 471 \times 4$ ($z \times y \times x \times ch$), where the last dimension is the quaternion component. Analogously, the Ti-7Al shown in 7(b) and 7(c) are of size $232 \times 674 \times 770 \times 4$ ($z \times y \times x \times ch$) and $224 \times 770 \times 770 \times 4$ ($z \times y \times x \times ch$) pixels respectively, with all edges cropped to produce a perfect parallelepiped volume. Each voxel in the Ti-6Al-4V set has resolution of $1.5 \times 1.5 \times 1.5 \mu\text{m}$, and in both Ti-7Al sets, each voxel has a resolution of $1.3 \times 1.3 \times 1.3 \mu\text{m}$.

These titanium alloys are composed primarily of the hexagonal close packed grains. In total, the Ti-6Al-4V dataset contains about 57,000 grains, visible in the IPF maps as regions of different color. The Ti-7Al material has larger grain size, with 500-1000 grains in each dataset. The datasets were proportionally divided into training, validation, and test subsets in ratios of 65%, 15%, and 20% respectively.

Sparsely Sectioned Input EBSD Data: Sparsely sectioned EBSD data are downsampled versions of the high-resolution EBSD data. However, because of how EBSD information is gathered, these EBSD data are not downsampled using pixel averaging. Instead sparsely sectioned EBSD data are produced by removal of xy planes in the z direction from the high-resolution ground truth with a downscale factor of 4x and 2x ($LR = \frac{1}{4}HR$ or $LR = \frac{1}{2}HR$). This is done to imitate the beam raster steps that would occur in an EBSD experiment with sparsely sectioned EBSD data, which would not influence the electron beam-material interaction volume at each location, but rather lead to greater raster distance between

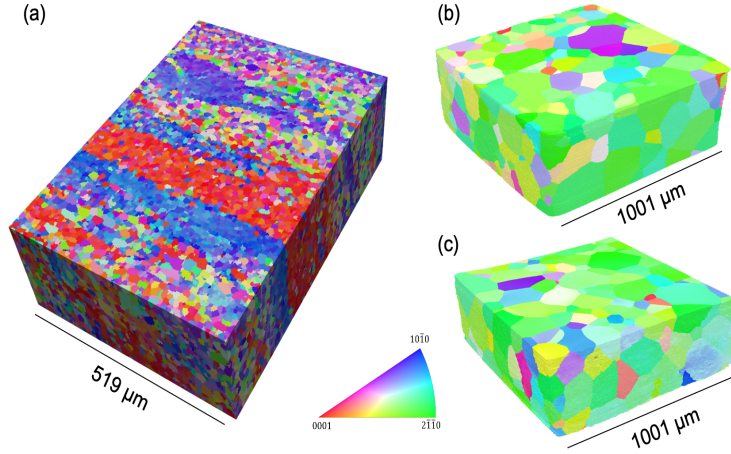


Figure 7: RENDERING OF 3D EBSD DATASET INVESTIGATED: shown in IPF coloring of the titanium alloys, (a) Ti-6Al-4V and (b) Ti-7Al mechanically loaded in tension to 1% strain and (c) 3%, used for network training, testing and validation. Dataset details are available elsewhere [45, 46].

consecutive locations.

10.3 Ablation Study

To show the effectiveness of quaternion convolution, we have done experiments with an enhanced deep scale residual (EDSR) network [47] and a quaternion enhanced deep residual (QEDSR) network as shown in figure 8. The QEDSR is similar to EDSR except that the real-convolution layer is replaced by quaternion convolution layer that helps in reducing computational complexity. With decreased complexity due to the quaternion convolution layer, we can add a quaternion transformer block into our network architecture to learn global features of EBSD maps. The PSNR/SSIM values for the generated EBSD maps and number of trainable parameters of different neural network architectures are shown in table 2. Our network QRBSA is giving better PSNR and SSIM than EDSR [47], while having fewer trainable parameters. To understand the effect of the quaternion transformer block in QRBSA, we have trained our network with and without the quaternion transformer block as shown in table 3. We see an improvement in PSNR and SSIM with the quaternion transformer block for the Ti6-Al-4V dataset, which has more crystallographic texturing and

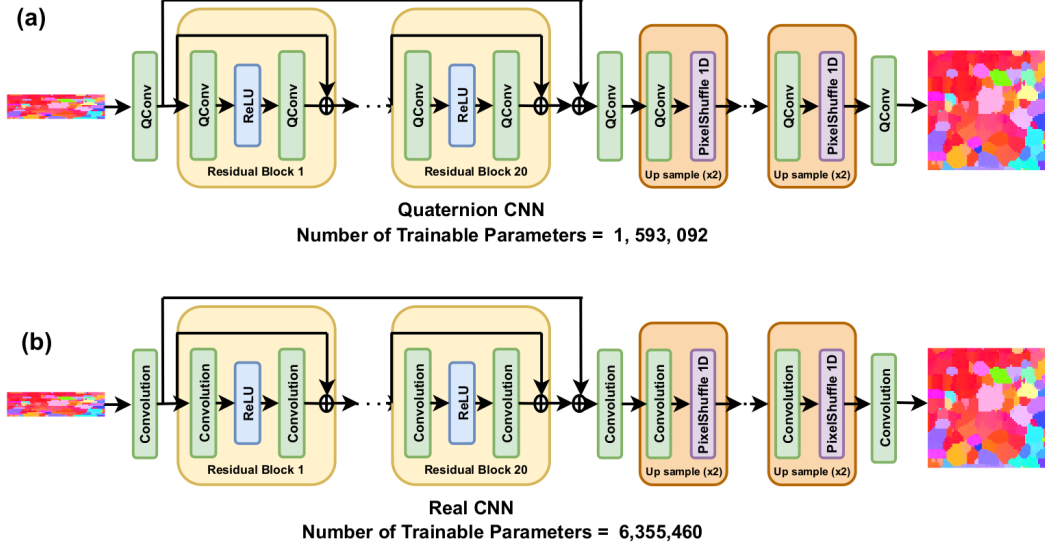


Figure 8: QUATERNION CNN AND REAL CNN: Both QCNN and RCNN have same number of layers except basic convolution operation layer. In QCNN, total number of trainable parameters are reduced significantly which give us room to add more complexity such self-attention layer to learn global features.

is difficult to learn because of the mismatched lengthscales over which it persists, compared to the grain structure. The quaternion transformer block will become even more useful for learning global features as we increase our library of 3D datasets available to our deep learning framework in future.

During analysis of our results, we found that some of generated xy planes (z_{normal}) in all three of the investigated datasets (Ti-6Al-4V, Ti-7Al 1 %, and Ti-7Al 3 %) have minor noise, as shown in [9](#). In our future work, we will be focusing on reducing this noise in the estimated xy planes by using 3D neural network architecture.

Network	Trainable Parameters	Ti-6Al-4V	Ti-7Al 1%	Ti-7Al 3%
HAN	63,315,578	21.21/ 0.839	29.36/ 0.919	31.26/ 0.941
EDSR	6,355,460	21.49/ 0.86	30.06/ 0.944	32.16/ 0.958
QEDSR	1,593,092	21.44/ 0.861	29.89/ 0.935	32.05/ 0.95
QRBSA	5,952,782	21.60/ 0.870	30.20/ 0.946	32.36/ 0.96

Table 2: **PSNR/SSIM:** COMPARISON OF PSNR/SSIM AND COMPLEXITY FOR DIFFERENT NETWORKS FOR SCALE FACTOR 2: Columns represent number of trainable parameters and PSNR/SSIM for different titanium datasets. A larger number is desired for both PSNR/SSIM.

Experiments	Ti-6Al-4V	Ti-7Al 1%	Ti-7Al 3%
QRBSA without quat transformer	18.12/ 0.718	27.27/ 0.906	29.49/ 0.9314
QRBSA with quat transformer	18.2/ 0.73	27.48/ 0.908	29.65/ 0.94

Table 3: **PSNR/SSIM:** COMPARISON OF PSNR/SSIM WITH AND WITHOUT QUATERNION TRANSFORMER FOR SCALE FACTOR 4: Columns represent PSNR/SSIM for different titanium datasets. A larger number is desired for both PSNR/SSIM.

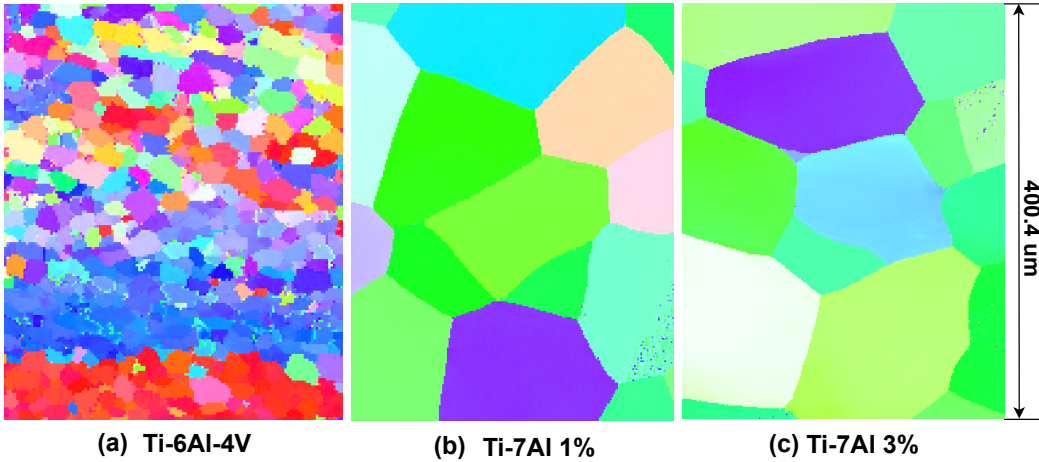


Figure 9: NOISE IN ESTIMATED XY PLANE: The deep learning framework is able to estimate the missing xy planes in z dimension but there are some minor noises in some of the xy planes.