Augmented Equivariant Attention Networks for Microscopy Image Transformation

Yaochen Xie, Yu Ding, Shuiwang Ji, *Senior Member, IEEE,*

***Abstract*— It is time-consuming and expensive to take high-quality or high-resolution electron microscopy (EM) and fluorescence microscopy (FM) images. Taking these images could be even invasive to samples and may damage certain subtleties in the samples after long or intense exposures, often necessary for achieving high-quality or high-resolution in the first place. Advances in deep learning enable us to perform various types of microscopy image-to-image transformation tasks such as image denoising, super-resolution, and segmentation that computationally produce high-quality images from the physically acquired low-quality ones. When training image-to-image transformation models on pairs of experimentally acquired microscopy images, prior models suffer from performance loss due to their inability to capture inter-image dependencies and common features shared among images. Existing methods that take advantage of shared features in image classification tasks cannot be properly applied to image transformation tasks because they fail to preserve the equivariance property under spatial permutations, something essential in image-to-image transformation. To address these limitations, we propose the augmented equivariant attention networks (AEANets) with better capability to capture inter-image dependencies, while preserving the equivariance property. The proposed AEANets captures inter-image dependencies and shared features via two augmentations on the attention mechanism, which are the shared references and the batch-aware attention during training. We theoretically derive the equivariance property of the proposed augmented attention model and experimentally demonstrate its consistent superiority in both quantitative and visual results over the baseline methods.**

***Index Terms*— Deep learning, attention networks, equivariance, microscopy images, image transformation, deep denoising, super-resolution, image transformation**

I. Introduction

Microscopy images of high quality in terms of resolution or noise level are desired to conduct research in various fields such as biomedical science and nanomaterial. However, the capture of high-quality microscopy images is usually at a high cost in budget and time, and may be infeasible under certain circumstances. This is especially critical for the observation of temporal dynamic or processes of live cells and organics, where exposure time and intensity enable more precise imaging but may reduce the temporal resolution and be harmful to the live cells [1]. To overcome these drawbacks but yet to produce higher quality images, studies such as microscopy image super-resolution or denoising, aims at computationally producing high-quality microscopy images from the physically acquired low-quality images.

Deep learning approaches consider the microscopy image super-resolution and denoising as image-to-image transformation tasks, in the sense that pairs of images, of the same size but different noise levels or resolutions, are used to train a deep neural network. One can then apply the trained deep neural network to the low-quality images for predicting their high-quality counterparts. Deep learning approaches have shown success in microscopy image transformation applications on both electron microscopy (EM) and fluorescence microscopy (FM) images, such as content-aware denosing [1], virtual refocusing [2], and super-resolution [3]–[5]. In particular, deep learning approaches that involve the self-attention mechanism [6] achieve even more promising performance on microscopy image transformation tasks, benefitting from its capability to perform non-local information aggregations and capture long-range dependencies [7].

Recently, several studies [8]–[10] have shown or indicated the importance of capturing the inter-images dependencies and shared features among images, due to the uniqueness of microscopy images. For example, [8] contemplate the question of what training strategy is the best for their EM super-resolution model and compare two strategies. The pooled-training uses all training image-pairs to train a single model and perform prediction on all testing areas, whereas the self-training trains a dedicated model for each unique testing area with the training image-pair from the same samples of the corresponding training area. For each model trained, the self-training uses far fewer image pairs than those used in pooled-training, yet [8] showed that the models trained with self-training generally yield higher performance than pooled-training, which on the surface appears to contradict the conventional wisdom in deep learning that a model trained with more data should generally perform no worse than that with fewer data. We attribute the reduced performance of pooled-training to the models' inability to fully utilize the additional information provided in

Manuscript received xxx; revised xxx. This work is partially supported by AFOSR DDIP program grant FA9550-18-1-0144, NSF grants IIS-1849085, IIS-1908220, and DBI-2028361, and Texas A&M X-grant program.

Y. Xie is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA, e-mail:(ethanycx@tamu.edu).

Y. Ding is with the Wm Michael Barnes'64 Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, USA, e-mail: (yuding@tamu.edu).

S. Ji is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA, e-mail: (sji@tamu.edu).

pooled-training. More importantly, we observe that this lack of capability is not unique to the deep learning methods tested by [8], but rather common among most existing deep learning methods. The methods are inadequate in terms of capturing inter-image dependencies and shared features among training image-pairs, which are prerequisite for ensuring the pooled-training strategy to do better.

To address the above issue, an existing attention-based approach has been proposed to endow neural networks with the capability to capture shared features among training instances. That is, to include the attention mechanism with learnable query as an augmentation to the original self-attention mechanism. Such an approach has been explored and commonly used in natural language processing (NLP) [11] and graph neural networks (GNNs) [12]. In the image domain, [9], [10] have also attempted to include the attention mechanism with learnable query to capture the inter-image dependencies and the common features shared among images.

However, we argue that the attention mechanism with learnable query can be inappropriate in the image-to-image transformation tasks, leading to potential performance reduction of the model due to the lacking of an essential property, the spatially permutation equivariant. When we perform spatial permutation such as rotation to an input image, the output image is desired to be permuted accordingly. For typical convolution-based deep models, such an equivariant property can be naturally learned or enforced by performing data augmentation such as rotation and flipping. However, involving the attention mechanism with learnable query makes such a property unsatisfied in an image-to-image transformation model and unable to be learned, unless constant values are output by the attention operator at all spatial locations. Consequently, although shared features can be captured, image-to-image transformation models involving the attention mechanism with learnable query suffers from performance loss due to the lack of permutation equivariance.

Motivated by both the desire to utilize inter-image dependencies and overcome the limitations of attention mechanisms with learnable query, we propose the augmented attention models with two components, the attention mechanism with shared references and the batch-aware attention applied in training. The resulting new attention model is referred to as the Augmented Equivariant Attention Networks (AEANets), whose attention block preserves the equivariance to any spatial permutations and can capture the inter-image dependencies and common features among images. We conduct experiments to evaluate the performance and effectiveness of the proposed AEANets. Quantitative results show that our AEANets significantly outperform the baselines on three microscopy image transformation tasks, i.e., super-resolution, denoising, projection, and segmentation, for both various types of biomedical images. We also demonstrate visually that AEANets produce better 3D-to-2D projection and super-resolution images compared to the respective baseline methods.

II. Preliminaries and Related Studies

In this section, we introduce the self-attention mechanism and related studies that apply the self-attention mechanism or its variations with learned query.

A. The Self-Attention Mechanism

The self-attention mechanism [6] has been widely applied to deep learning models in natural language processing (NLP) [13] and computer vision [14]. Compared to local operations such as convolutions that can only aggregate information locally, the self-attention mechanism is able to incorporate global information. Given an input feature map, the self-attention mechanism computes the relevance between every two locations on the feature map and aggregations information from one location to another according to the relevance. The self-attention mechanism hence endows neural networks the capability to capture long-range dependencies.

The self-attention mechanism can be applied to feature maps $\mathcal{X} \in \mathbb{R}^{s_1 \times \cdots \times s_k \times c}$ with any $k \geq 1$ where k denotes the number of spatial dimensions, s_i denotes the spatial size along the i-th dimension and c denotes the number of features. For example, in a 2D image case, the self-attention mechanism is applied on the input $\mathcal{X} \in \mathbb{R}^{w \times h \times c}$ where w and h denote the width and height of the image. Without loss of generality, we describe how the self-attention operator is performed in the 1D case ($k = 1$), where there is only one spatial dimension. For higher-dimensional cases, the spatial dimensions can be unfolded into one dimension $s = s_1 s_2 \cdots s_k$ before being given to the self-attention operator. The output of the self-attention operator can then be folded back to the original shape as the final output.

In the 1D case, the self-attention operator takes as input a matrix $\boldsymbol{X} \in \mathbb{R}^{s \times c}$ representing the features of a sequence, where s denotes its spatial dimension (*i.e.*, the length of the sequence or the spatial dimensions of unfolded images) and c denotes its feature dimension. The self-attention operator firstly computes three matrices, *i.e.*, the query $\boldsymbol{Q}$, the key $\boldsymbol{K}$ and the value $\boldsymbol{V}$, by performing convolutions with kernel size of 1, to the input matrix $\boldsymbol{X}$. Formally, $\boldsymbol{Q} = q(\boldsymbol{X}) \in \mathbb{R}^{s \times c_1}$, $\boldsymbol{K} = k(\boldsymbol{X}) \in \mathbb{R}^{s \times c_1}$ and $\boldsymbol{V} = v(\boldsymbol{X}) \in \mathbb{R}^{s \times c_2}$, where $q(\cdot), k(\cdot), v(\cdot)$ are three independent projections. Then, the output $\boldsymbol{Y}$ of the attention operator is computed by

$$\boldsymbol{Y} = \text{Normalize}(\boldsymbol{Q} \cdot \boldsymbol{K}^T) \cdot \boldsymbol{V} \in \mathbb{R}^{s \times c_2}. \tag{1}$$

The function Normalize$(\cdot)$ performs a normalization on the attention map $\boldsymbol{Q} \cdot \boldsymbol{K}^T$ so that the values on the output $\boldsymbol{Y}$ will not scale with the spatial size. Commonly used Normalize$(\cdot)$ functions includes Softmax$(\cdot)$ and the division by the spatial size of $\boldsymbol{K}$, *i.e.*,

$$\text{Normalize}(\boldsymbol{Q} \cdot \boldsymbol{K}^T) = \frac{1}{s}(\boldsymbol{Q} \cdot \boldsymbol{K}^T). \tag{2}$$

In this work, we use the normalization function in Equation (2). For clear comparisons, we use this type of normalization when describing all variations of the self-attention mechanisms in the rest of the paper.

Note that although the spatial sizes of $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are the same in the self-attention mechanism, the spatial size s of the output is determined by the spatial size of query $\boldsymbol{Q}$. In addition, the feature size c_2 of the output is determined by the feature size of value $\boldsymbol{V}$.

B. Attention Mechanism with Learned Query

A common variation of the attention mechanism is to directly learn the values in the query matrix $\boldsymbol{Q}$. In this case, the query $\boldsymbol{Q}$ does not depend on the input $\boldsymbol{X}$. The attention mechanism with a learnable query is commonly used in NLP [11] and graph neural networks (GNNs) [12]. In certain domains such as biomedical image and nanoparticles, different images from one dataset usually share similar patterns and common features, such as microscopy images captured from different parts of tissue or tissues of the same type. The power of the learnable query has also been explored by previous studies in the biomedical image domain [9], [10]. In these cases, such a variation of the attention mechanism allows the networks to capture common features from all input images during training since the query is independent of the input and is shared by all input images.

Instead of computing the query from $\boldsymbol{X}$, the attention mechanism can learn a matrix $\boldsymbol{Q} \in \mathbb{R}^{s_q \times c_1}$ independently of $\boldsymbol{X}$. The other computations in the attention mechanism with learnable query is then the same as the self-attention mechanism. Formally, with $\boldsymbol{K} = K(\boldsymbol{X}) \in \mathbb{R}^{s \times c_1}$, $\boldsymbol{V} = V(\boldsymbol{X}) \in \mathbb{R}^{s \times c_2}$ and a directly learned matrix $\boldsymbol{Q} \in \mathbb{R}^{s_q \times c_1}$,

$$\boldsymbol{Y} = \text{Normalize}(\boldsymbol{Q} \cdot \boldsymbol{K}^T) \cdot \boldsymbol{V} \in \mathbb{R}^{s_q \times c_2}. \tag{3}$$

Since the spatial size of the output $\boldsymbol{Y}$ is determined by the spatial size of $\boldsymbol{Q}$, the output size of the attention mechanism with learned query is no longer related to the spatial size of $\boldsymbol{X}$. As a result, when the attention mechanism with learned query is included in the neural network, the size of the output of the network is usually fixed.

III. Model Augmentation with Shared References

Previous studies [9], [10] have explored different approaches to include a learnable query in the attention operator to capture common features among different images. Such attention operators with learnable query have been shown to bring a promising performance boost, especially in NLP and image classification tasks. However, the attention operators with learnable query can, on the contrary, limit the performance of models for image-to-image transformation tasks such as image super-resolution, as such operators are not able to preserve an essential property required by the image-to-image transformation models, *i.e.*, the equivariance to spatial permutations. Note that such properties are not changed by increasing network depth, modifying architectures other then the attention operator, or simply applying data augmentations. Therefore, the issue cannot be addressed by these approaches.

In this section, we analyze the equivariance property in subsection III-A, and show that such property is violated when the attention mechanism includes learned query in subsection III-A. Based on our analysis, we propose in subsection III-C the attention operator with shared references that are able to capture common features among images without violating the equivariance property.

A. Equivariance and Invariance to Spatial Permutations

The spatial permutation includes a group of transformations to be applied to images. It is performed by permuting the spatial locations of any number of pixels or voxels in an image. Some common examples of spatial permutation include the rotation, the flipping and the shifting of an image. The equivariance to the spatial permutation is a property of an operator or a model such that applying a spatial permutation to the input of the operator or the model results in an equivalent effect of applying the same spatial permutation to the output. On the contrary, if an operator is invariant to spatial permutations, then its output remains unchanged when permuting the input. We provide formal definitions of the spatial permutation, the equivariance and invariance property below.

Definition 1: Consider an image or feature map $\boldsymbol{X} \in \mathbb{R}^{s \times c}$, where s denotes the spatial dimension and c denotes the number of features. Let π denote a permutation of s elements. We call a transformation $\mathcal{T}_\pi : \mathbb{R}^{s \times c} \rightarrow \mathbb{R}^{s \times c}$ a spatial permutation if $\mathcal{T}_\pi(\boldsymbol{X}) = P_\pi \boldsymbol{X}$, where $P_\pi \in \mathbb{R}^{s \times s}$ denotes the permutation matrix associated with π, defined as $P_\pi = \left[\boldsymbol{e}_{\pi(1)}, \boldsymbol{e}_{\pi(2)}, \cdots, \boldsymbol{e}_{\pi(s)}\right]^T$, and $\boldsymbol{e}_i$ is a one-hot vector of length s with its i-th element being 1.

Definition 2: We call an operator $A : \mathbb{R}^{s \times c_1} \rightarrow \mathbb{R}^{s \times c_2}$ to be spatially permutation equivariant if $\mathcal{T}_\pi(A(\boldsymbol{X})) = A(\mathcal{T}_\pi(\boldsymbol{X}))$ for any X and any spatial permutation $\mathcal{T}_\pi$. In addition, an operator $A : \mathbb{R}^{s \times c_1} \rightarrow \mathbb{R}^{s \times c_2}$ is spatially permutation invariant if $A(\mathcal{T}_\pi(\boldsymbol{X})) = A(\boldsymbol{X})$ for any X and any spatial permutation $\mathcal{T}_\pi$.

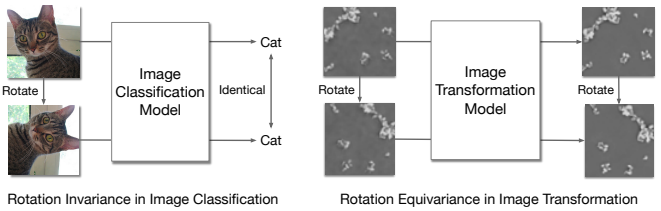

Fig. 1. Examples that compare the invariance property with the equivariance property. Tasks such as classification require spatial permutation (e.g., rotation) invariant models, where applying the permutation to the input does not change the output of the model. On the contrary, the image transformation tasks require spatial permutation (e.g., rotation) equivariant models, where applying the permutation to the input leads to the same permutation applied to the output of the model.

We argue that while the image classification models could benefit from the invariance to spatial permutations according to previous studies [15], [16], and an image-to-image transformation model requires the equivariance to spatial permutations, as shown in Figure 1. Detailed discussions about the properties are provided in Appendix I.

B. Spatial Permutation Properties of Attention Operators

We now analyze the properties of the two types of attention operators regarding the spatial permutation using the same notations as in Section II-A and Section II-B. Intuitively, when a spatial permutation is performed on the input of an attention operator, the corresponding permutation made on the key $\boldsymbol{K}$ and the value $\boldsymbol{V}$ does not result in any difference in the output,

as long as the same permutation is applied to both $\boldsymbol{K}$ and $\boldsymbol{V}$. In fact, the order of spatial locations on the output feature map is determined by the spatial locations on the query $\boldsymbol{Q}$. Hence the attention operator is permutation equivariant as long as $\boldsymbol{Q}$ is obtained from $\boldsymbol{X}$.

For simplicity, we denote A_s a self-attention operator and $A_{\boldsymbol{Q}}$ an attention operator with learned query. The outputs $\boldsymbol{Y}$ of the two operators are therefore equal to $A_s(\boldsymbol{X})$ and $A_{\boldsymbol{Q}}(\boldsymbol{X})$. We show that the following theorem holds.

Theorem 1: A self-attention operator A_s is permutation equivariant while an attention operator with learned query $A_{\boldsymbol{Q}}$ is permutation invariant. In particular, letting $\boldsymbol{X}$ denote the input matrix and $\mathcal{T}$ denotes any spatial permutation, we have

$$A_s(\mathcal{T}_\pi(\boldsymbol{X})) = \mathcal{T}_\pi(A_s(\boldsymbol{X})),$$

and

$$A_{\boldsymbol{Q}}(\mathcal{T}_\pi(\boldsymbol{X})) = A_{\boldsymbol{Q}}(\boldsymbol{X}).$$

The proof of Theorem 1 is provided in Appendix II. Note that we prove the above theorem with the normalization of division by spatial size of $\boldsymbol{K}$. The theorem still holds when the Softmax is applied for normalization, the proof of which can be found in [17].

C. Augmented Attention with Shared References

We have shown that an image-to-image transformation model requires equivariance to spatial permutations while the attention operator with learned query is permutation invariant. Although the attention operator with learned query endows models the capability to capture common features among images, the invariance property makes it inappropriate to be applied in image-to-image transformation tasks. In order to endow the attention operator the capability to capture common features among images without losing the equivariance property, we propose an attention operator augmented with learnable shared references, as opposed to shared query. The shared references are represented by a matrix consisting of learnable variables and are augmented to the key and value matrices along the spatial dimensions.

To be concrete, given the flattened input feature map $\boldsymbol{X} \in \mathbb{R}^{wh \times c}$ where the width w and height h are flattened into one dimension and c is the number of features, the shared references are represented by a learnable matrix $\boldsymbol{R} \in \mathbb{R}^{r \times c}$, where r is the size of the shared references as a hyper-parameter. The learned shared references is projected by $k(\cdot)$ and $v(\cdot)$ into the same space of key and value. The computation of an attention operator augmented with shared references $A_{\boldsymbol{R}}$ can be formally expressed as

$$\begin{aligned} \tilde{\boldsymbol{X}} &= \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{X} \end{bmatrix} \in \mathbb{R}^{(r+wh)\times c}, \\ A_{\boldsymbol{R}}(\boldsymbol{X}) &= \frac{1}{r+wh}(q(\boldsymbol{X}) \cdot k^T(\tilde{\boldsymbol{X}})) \cdot v(\tilde{\boldsymbol{X}}), \end{aligned} \tag{4}$$

where $k^T(\tilde{\boldsymbol{X}})$ denotes the transposed key $k(\tilde{\boldsymbol{X}})$. Note that the key $\tilde{\boldsymbol{K}} = k(\tilde{\boldsymbol{X}})$ and value $\tilde{\boldsymbol{V}} = v(\tilde{\boldsymbol{X}})$ are computed from $\tilde{\boldsymbol{X}}$, while the query $\boldsymbol{Q}$ is computed from $\boldsymbol{X}$. The operator with shared references is illustrated in Figure 2. We now show the property of the proposed attention operator $A_{\boldsymbol{R}}$ in the following theorem.

Theorem 2: The proposed augmented attention operator with shared references $A_{\boldsymbol{R}}$ is spatially permutation equivariant, *i.e.*,

$$A_{\boldsymbol{R}}(\mathcal{T}_\pi(\boldsymbol{X})) = \mathcal{T}_\pi(A_{\boldsymbol{R}}(\boldsymbol{X})).$$

The proof of Theorem 2 is provided in Appendix III.

Compared with the original self-attention operators in which the key and value matrices are fully based on the input, the key and value of $A_{\boldsymbol{R}}$ contain additional information about the features shared by all images in the dataset. The learning process of the shared reference is to distill common features from images in the entire training data. Each spatial location on an input instance aggregates information not only globally from the input instance itself, but also from the distilled references shared by all the input images.

IV. Model Augmentation with Batch-Aware Training

A common strategy to train a deep model is to feed a mini-batch of images to the network at each training step. The mini-batch is referred to as batch in the following paragraphs for short. When a self-attention operator is included, the operator processes the batch at an instance level. Given an input batch of feature maps $\{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_N\}$, where N is the batch size, the self-attention operator computes the outputs individually for each instance in the batch, *i.e.*, $\boldsymbol{Y}_i = A_s(\boldsymbol{X}_i)$ for $i = 1, \cdots, N$.

In the case where images in a dataset share similar patterns, the performance of a deep model can further benefit by incorporating cross-images dependencies. In this case, the learning of such dependencies across images in a batch can be of great importance. Due to the non-local property, the attention operators can be extended from the instance level to a batch level in order to learn the correlations across images in a batch. Formally, we define an augmented batch-aware operator A_{batch} such that

$$\boldsymbol{Y}_i = A_{batch}(\boldsymbol{X}_i; \boldsymbol{X}_1, \cdots, \boldsymbol{X}_N), i = 1, \cdots, N.$$

In this case, the computation of each output instance is aware of the other instances in the current batch. In order to realize such an augmentation, we propose a training strategy with this batch-aware attention, where the key and value cover all the images in the training batch. That is,

$$\begin{aligned} &A_{batch}(\boldsymbol{X}_i; \boldsymbol{X}_1, \cdots, \boldsymbol{X}_N) = \\ &\frac{1}{Nwh} q(\boldsymbol{X}_i) \cdot k^T \left(\begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_N \end{bmatrix} \right) \cdot v \left(\begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_N \end{bmatrix} \right), \end{aligned} \tag{5}$$

where $k(\cdot)$ and $v(\cdot)$ are the projections defined in the original self-attention.

The proposed batch-aware attention aggregates information from the entire batch based on the correlation across images for each input location. Since each batch is uniformly sampled from the entire dataset, the aggregation from batches can estimate the information aggregation from the entire dataset. The weights of the projections $q(\cdot), k(\cdot), v(\cdot)$ in batch-aware

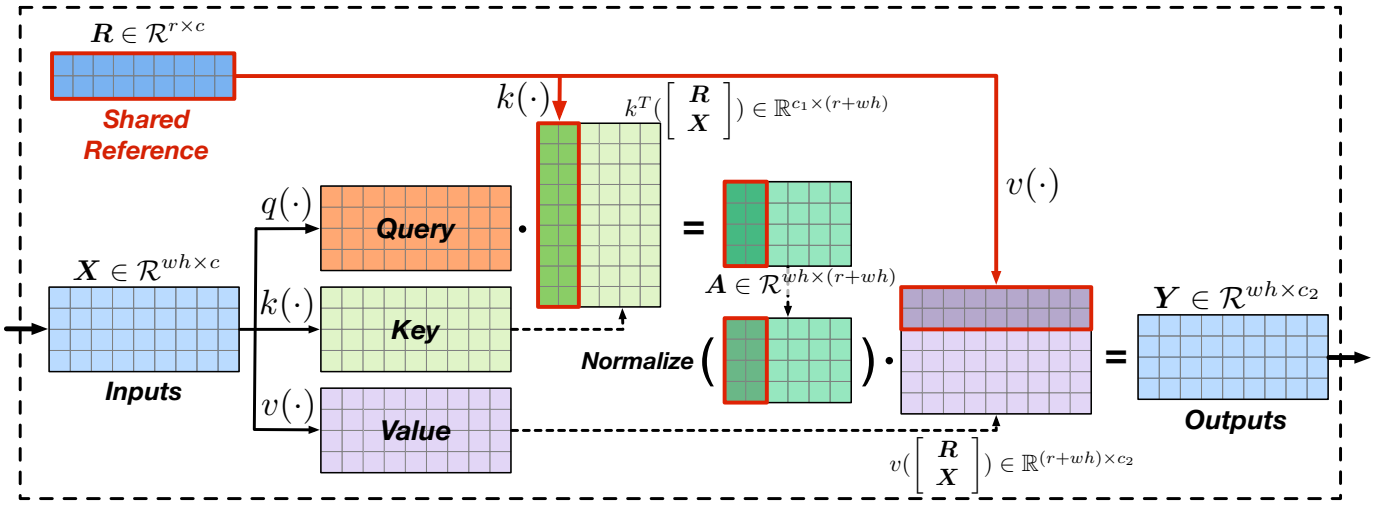

Fig. 2. Comparison between the original self-attention (top) and proposed attention operator with shared references (bottom). Given the flattened input matrix $\boldsymbol{X}$ of shape [width $\times$ height, channels], the proposed attention operator train a reference matrix of shape [reference size, channels], which is used to compute the key and value. The highlighted parts in the key and value are related to the shared references and the rest parts are related to the input image. The spatial size of the output is the same as the spatial size of the query.

attention and the attention with shared references are shared during training. The purpose of including the batch-aware attention in training is to help the distill of shared references and learn better projections in attention by using cross-image dependencies. In our experiments, we empirically show that additional performance improvements can be achieved by including the augmented batch-aware attention in training.

V. Augmented Equivariant Attention Networks

A. The Augmented Equivariant Attention Block

The proposed augmented equivariant attention block consists of a branch for the attention operator $A_{\boldsymbol{R}}$ with shared references and a branch for the batch-aware attention A_{batch}. In the attention block, the weights in each projection of $q(\cdot), k(\cdot)$ and $v(\cdot)$ are shared by the two attention operators $A_{\boldsymbol{R}}$ and A_{batch}. As the attention block performs differently during training and prediction, we individually describe how it works during training and prediction.

During training, both the $A_{\boldsymbol{R}}$ and A_{batch} are used. Given an input batch $\{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_N\}$ to the block, the batch is evenly split into two groups $\{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_{\lfloor N/2 \rfloor}\}$ and $\{\boldsymbol{X}_{\lfloor N/2 \rfloor + 1}, \cdots, \boldsymbol{X}_N\}$ as the inputs to the two branches. The outputs of the two branches are then merged back into a complete batch. While the batched data in the two branches are separate, the parameters of the two branches are shared.

Once the network is trained, the parameters in the two operators are fixed and are used by the attention operation with shared references during prediction. The batch-aware attention is excluded during prediction since there is not necessarily a batch input during prediction. In other words, the branch that contains the batch-aware attention operator is disabled, and all input data flow into the $A_{\boldsymbol{R}}$ branch. In spite of the exclusion of the batch-aware attention, the existence of shared references distilled from training images allows the model to still utilize the dependencies between training images and the given input image for prediction. Figure 3 illustrates how the block works during training and prediction.

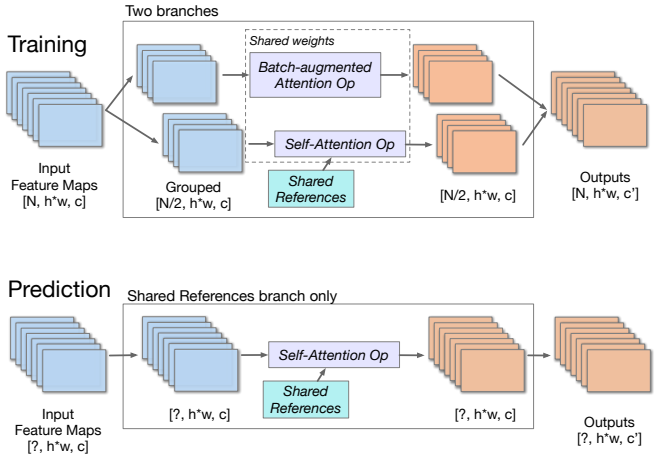

Fig. 3. The proposed attention block. During training (top), both branches are used. The input batch are splitted into two groups and passed to the two operators. The outputs of the two operators are then merged together. For prediction (bottom), only the branch with shared references is used.

B. Network Architecture

Recent studies have shown that the U-Net [18] architecture achieves promising performance in many image transformation tasks, especially for microscopy images [9], [19], [20]. In this work, we use the U-Net as the base network architecture of our model. To be specific, we use a U-Net with a depth of 3 (including two down-sampling operations and up-sampling operations, respectively). The skip-connections in the U-Net are merged into the up-sampling path by concatenation.

To enable the use of the proposed augmented attention within the U-Net architecture, we follow [7] to (1) include a residual connection in the attention block by addition and (2) substitute the bottom block in the original U-Net with our proposed attention block. In addition, the proposed attention blocks can be also applied as up-sampling blocks by performing up-sampling to the query $\boldsymbol{Q}$ [7]. The overall architecture

of the proposed network is shown in Appendix IV.

Note that although we use the augmented attention blocks in the U-Net architecture in this study, it can be inserted into any other deep architectures, thereby capturing non-local and cross-image dependencies and the common features shared by the entire dataset.

VI. Experimental Results

We evaluate the proposed AEANet for different microscopy image transformation tasks on three microscopy image datasets captured by different instruments. The datasets are the Paired Electron Microscopy (EM) Image Dataset [8], the Planaria dataset for 3D image denoising; and the Flywing dataset for 3D image projection—the latter two datasets are from CARE [1] and captured by fluorescence microscopy. For all the three datasets, the low-quality images and their high-quality counterparts are physically captured. For each of the three datasets, we follow baseline methods for their experimental settings including training-test split and basic neural network configurations for fair comparisons. For all microscopy image transformation tasks, we use two evaluation metrics, the structural similarity index measure (SSIM) and peak signal-to-noise ratio (PSNR), calculated between the prediction and the high-quality images (ground truth). We additionally conduct experiments on the 3D segmentation task with brain Magnetic Resonance Images (MRI) to demonstrate broader application scenarios for the proposed methods. We summarize the implementation details and configurations of our methods for individual experiments in Appendix V.

A. The Paired EM Image Dataset

We first train and evaluate our model on the publicly available[1] paired EM images dataset [8]. The paired image dataset consists of 22 pairs of LR and HR nanoimages of size $1,280 \times 944$. The LR and HR nanoimages are captured by the same scanning electron microscope (SEM) at two different magnifications. Specially, the HR image is two times zoomed-in from the LR image and the field of view (FOV) of the HR image is covered by the FOV of the LR image, *i.e.*, the HR image corresponds to a $1/4$ sub-area of the LR image. Several preprocessing steps are performed on the original LR and HR images to build our dataset for training and testing. We first perform the Random Sample Consensus [21] (RANSAC) algorithm to register the HR images in the corresponding areas in the LR images, based on the ORB features [22]. We select the registered area in each LR image and use a bicubic interpolation to upsample the selected LR subareas to be of the same size of the HR images, *i.e.*, $1,280 \times 944$. Through the preprocessing, the resulting LR and HR images refer to the same area but are in different resolutions.

1) Training-Test Split and Training Strategies: As described in Section I, [8] studied two training strategies, self-training and pooled-training, for the EM image super-resolution. While both self-training and pooled-training are practical in real scenarios of super-resolution, we only focus on the pooled-training, which is more common in machine learning studies, in our evaluation. In particular, the pooled-training trains a single model on all the 22 image pairs. With the splitting of the original images into 3×4 smaller sub-images, in the pooled-training, the 22 image pairs become a total of 198 sub-image pairs for training and 66 sub-image pairs for test. The evaluation metrics are computed on each testing sub-image and then averaged.

2) Evaluation Results: We evaluate our method and compare it with an array of deep learning-based baselines. In addition to three deep learning methods compared in [8], *i.e.*, VDSR [23], RCAN [24], and EDSR [25], we further include the original U-Net [18] and GVTNets [7], the current state-of-the-art model for microscopy image transformation, in this comparison study. The U-Net, GVTNets, and our proposed AEANets use the same network architecture setting except for the attention blocks. We also include the SOTA non-deep-learning-based method, which is the paired LB-NLM [8] in our baselines. We show in Table I the averaged improvements in terms of the two metrics, as compared to the input LR image (i.e., after bicubic interpolation). The improvements in the two metrics are denoted as ΔPSNR and ΔSSIM.

The results show that the AEANets model with pooled-training significantly outperforms the baselines with the same training strategy. More importantly, the performance of AEANets with pooled-training is better than the baseline models with self-training, indicating that the self-training is no longer required for our proposed AEANets. In other words, AEANets can be used more efficiently, leading to better performance, and can be applied in broader scenarios where self-training may not be applicable.

While the improvement in terms of PNSR is moderate, the improvement in terms of SSIM is much more remarkable, a 70% increase as compared to the best of the three deep learning methods originally used in [8]. Recall that SSIM measures how far away an image is from the HR image and a higher SSIM suggests a better capability of the resulting image to show finer details. The improvement in SSIM bears important practical implication for material characterization.

To see the implication from a different angle, consider the image quality improvement only in the foreground of the images. Not surprisingly, material scientists are more interested in the nanomaterial clusters (foreground) than the host material (background) in their applications. To separate the foreground and background, we follow [8] and perform Otsu's algorithm on each testing patch. We compute the improvements in PSNR on the foreground and background for the following methods: VDSR, SRSW, Paired LB-NLM and AEANets. The outcome is shown in Table I (right). The results demonstrate that although VDSR with self-training achieves a higher PSNR in the background, AEANets outperform the three aforementioned methods for the foreground. This outcome reinforces the advantage of AEANets shown in the SSIM comparison.

In terms of the foreground-background difference, [8] in fact commented that "*It is apparent that all these methods [those included in their paper] denoise the background much*

[1]The paired EM images dataset is available for public access at https://aml.engr.tamu.edu/2001/09/01/publications/ (then go to J74).

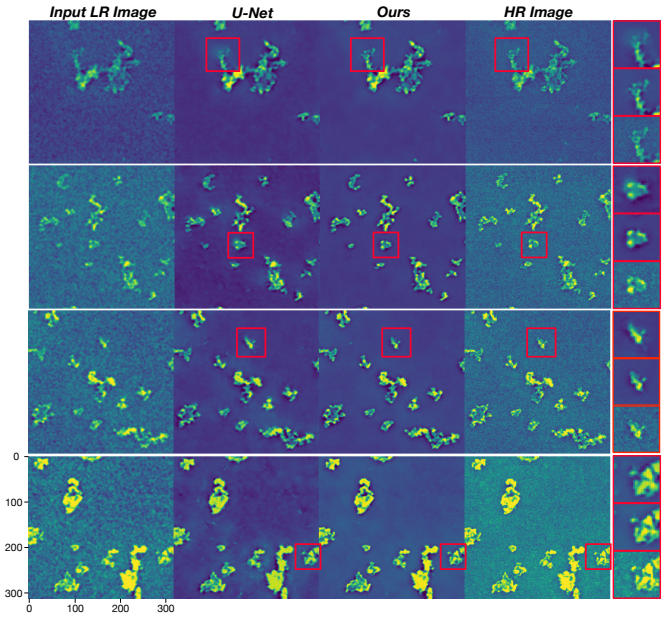

Fig. 4. Visualization of the output of super-resolution models on four testing patches. From left to right, the columns are input LR images, outputs of the U-Net, outputs of our model and the ground truth HR images). We select some areas to zoom in for a better view. From top to bottom, smaller patches on the right are zoomed-in from U-Net, our model and the HR images.

more than they enhance the foreground." While AEANets still see a greater PSNR improvement over its background, its foreground-background performance gap is the smallest among the four alternatives in Table I (right). The practical implication is that AEANets are a better tool for material characterization.

We also include some visualization results to compare the super-resolution performance of AEANets with that of U-Net. The visualization shows that the predictions from AEANets have clearer edges of the nanomaterial clusters. It is also worth mentioning that the deep learning-based methods, including U-Net and AEANets, have an additional denoising effect in the background. We believe that the effect is due to the noise in the LR image and HR image belongs to the same noise distribution but are independently sampled, which satisfies the requirements of the Noise2Noise [26] denoising. The denoising effect could be less likely to occur in synthetic super-resolution datasets. Compared to U-Net, AEANets can perform better denoising, as shown in Figure 4.

B. The 3D FM Image Datasets

We further evaluate our methods on Planaria and Flywing, two 3D microscopy transformation datasets from CARE [1]. The Planaria dataset evaluates the model performance on the content-aware 3D image denoising task. It includes 17,005 pairs of noisy-clean 3D image patches of shape $64\times64\times16$ for training. The test data consists of 20 larger 3D images of size $1024\times1024\times95$. For each test image, the ground truth image and noisy images at three different noise levels captured under different lighting conditions (C1, C2, and C3) are provided. We evaluate our model on all the three noise levels. The Flywing dataset evaluates the model performance on the content-aware 3D-to-2D image projection. Given a noisy 3D image, the projection task requires the transformation model to predict the surface of an organism in the 3D image and project it into a 2D images, excluding the noise along the depth dimension. The Flywing dataset consists of 16,891 pairs of noisy 3D and clean 2D patches for training and 26 test images. Similar to the Planaria dataset, each test image contains the ground truth version and the noisy version at three different noise levels.

Following the baseline configuration [7], we apply our augmented attention operators to both the bottom block and up-sampling blocks for CARE datasets. For the 3D-to-2D projection task, we follow [1], [7] for the base model consisting of a 3D U-Net for surface projection followed by a 2D U-Net that further performs denoising on the projected image. Augmented attention blocks are included in both 3D and 2D U-Nets. As the 3D images for training are already large enough in their spatial size due to the additional depth dimension, we only apply the shared references and omit the batch-aware attention during training to avoid memory issue. In particular, the computational cost of an attention operator for a 3D input of spatial size $w \times h \times d$ can be $O(d^2)$ times the cost of a 2D input of spatial size $w \times h$.

The evaluation results in terms of PSNR and SSIM are shown in Table II and Table III. We include the evaluation metrics computed on the input images, and the predictions of the baseline methods, the U-Net and the GVTNet. We also include visualizations of the prediction results of three methods on the Flywing dataset in Figure 5. The shown predictions are performed on noisy images with the worst lighting condition (C3). Both quantitative and visual results show that the proposed AEANet further consistently outperforms the current state-of-the-art methods on a wider range of microscopy image transformation tasks, indicating the effectiveness of the proposed augmented attention blocks.

C. 3D Brain MRI Segmentation

To demonstrate the effectiveness of AEANets on medical images and additional tasks, we further perform the evaluation on the 3D multimodality isointense infant brain MR image (MRI) dataset [27]. The MRI segmentation task aims to segment given MR images by identifying different regions including cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM) regions. The MRI segmentation is also considered as an image transformation task as it performs pixel-wise classification and hence requires the model to be spatially permutation equivariant.

We follow [28] for the network, training configuration, and evaluation setting only except for the AEA block. In particular, we perform the leave-one-subject-out cross-validation on the ten public MRI subjects and compute the Dice ratio as the evaluation metric. The results shown in Table IV indicate that AEANets achieve consistently better performance compared to the close baseline Non-local U-Net.

D. Verification of Equivariance Properties

To better support our theroy and claims, we empirically verify the spatial permutation equivariance property of AEANets

TABLE I

Left: COMPARISON OF PERFORMANCE QUANTIFIED BY IMAGE QUALITY IMPROVEMENT IN TERMS OF PSNR AND SSIM, AMONG OUR METHODS AND THE BASELINE MODELS. BOLD NUMBERS ARE THE HIGHEST COMPARED AMONG RESULTS IN BOTH TRAINING STRATEGIES. *Right*: IMPROVEMENTS IN PSNR ON FOREGROUND (NANOMATERIAL CLUSTERS) AND BACKGROUND (HOST MATERIAL) INDIVIDUALLY. "SELF" IN THE BRACKETS AFTER METHOD NAMES REFERS TO SELF-TRAINING.

	Pooled-training		Self-training	
Methods	ΔPSNR (dB)	ΔSSIM	ΔPSNR (dB)	ΔSSIM
VDSR [23]	1.25	0.047	2.07	0.051
RCAN [24]	1.59	0.051	2.07	0.050
EDSR [25]	1.35	0.051	2.06	0.052
Paired LB-NLM [8]	0.78	0.031	1.67	0.037
U-Net [18]	1.46	0.074	–	–
GVTNet [7]	1.87	0.086	–	–
Learned Query	1.64	0.084	–	–
AEANet (Ours)	**2.10**	**0.087**	–	–

Methods	Foreground	Background
VDSR (Self)	0.97	**2.83**
SRSW (Self)	-0.25	2.15
Paired LB-NLM (Self)	0.23	2.65
Learned Query (Pooled)	0.75	1.95
AEANet (Pooled)	**1.15**	2.42

TABLE II

EVALUATION RESULTS ON THE PLANARIA DATASET FOR 3D IMAGE DENOISING. FOR ALL THE THREE NOISE LEVELS (C1, C2, AND C3), MODEL PERFORMANCE IN TERMS OF SSIM AND PSNR ARE PROVIDED. THE STANDARD ERRORS ARE COMPUTED AMONG TEST SAMPLES FOLLOWING PREVIOUS STUDIES. THE AVERAGED SCORES OVER THE THREE LEVELS ARE ALSO PROVIDED.

	C3 (SSIM)	C3 (PSNR)	C2 (SSIM)	C2 (PSNR)	C1 (SSIM)	C1 (PSNR)	Avg (SSIM)	Avg (PSNR)
Input	0.1561	21.43	0.1827	21.73	0.2260	22.22	0.1883	21.79
Unet	0.6441±0.1207	28.13±1.37	0.7397±0.0885	30.15±1.66	0.7707±0.0889	31.57±1.71	0.7182	29.95
GVTNet	0.6972±0.1177	28.63±1.42	0.7745±0.0886	30.88±1.65	0.7929±0.0824	31.95±1.64	0.7549	30.49
AEANet	**0.7073±0.0994**	**28.64±1.39**	**0.7764±0.0897**	**30.95±1.84**	**0.7933±0.0838**	**32.08±1.70**	**0.7590**	**30.56**

TABLE III

EVALUATION RESULTS ON THE FLYWING DATASET FOR 3D-TO-2D IMAGE PROJECTION. FOR ALL THE THREE NOISE LEVELS (C1, C2, AND C3), MODEL PERFORMANCE IN TERMS OF SSIM AND PSNR ARE PROVIDED. THE STANDARD ERRORS ARE COMPUTED AMONG TEST SAMPLES FOLLOWING PREVIOUS STUDIES. THE AVERAGED SCORES OVER THE THREE LEVELS ARE ALSO PROVIDED.

	C3 (SSIM)	C3 (PSNR)	C2 (SSIM)	C2 (PSNR)	C1 (SSIM)	C1 (PSNR)	Avg (SSIM)	Avg (PSNR)
Input	0.0241	16.62	0.0795	17.23	0.1902	18.38	0.0979	17.41
Unet	0.5592±0.0403	21.96±0.48	0.5971±0.0705	22.55±1.14	0.6067±0.0216	23.66±0.26	0.5877	22.72
GVTNet	0.5908±0.0465	22.36±0.43	0.6954±0.0248	24.28±0.38	0.7511±0.0257	25.81±0.33	0.6791	24.15
AEANet	**0.6008±0.0452**	**22.50±0.43**	**0.7074±0.0305**	**24.54±0.41**	**0.7600±0.0195**	**26.03±0.31**	**0.6894**	**24.36**

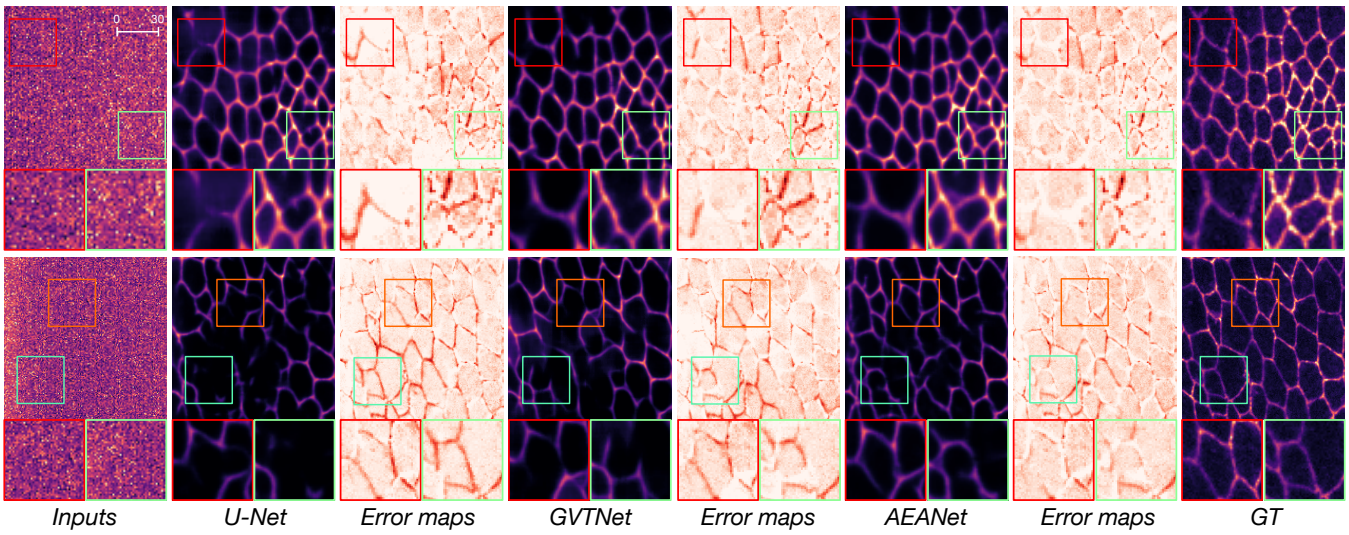

Fig. 5. Visualization of the predicted 2D surface projections from the Flywing dataset. From left to right, the columns are the projection from input noisy volume, the predictions and error maps of U-Net, GVTNet, AEANet (ours), respectively, and the ground truth images. The images are predicted from noisy images with the worst lighting condition (C3). We zoom in some subareas for a better view.

and attention with learned query. In particular, we visualize the first two channels of the query tensor and attention outputs when the raw patch and a rotated patch are input to the model. The visualizations are shown in Figure 6. For the attention block with learned query, the values in both query and attention output tensor reduce to nearly constant among spatial locations (according to the histograms) and do not rotate accordingly with the input patch. This is due to the permutation invariance nature of attention block with learned query, who becomes permutation equivariance if and

TABLE IV

EVALUATION RESULTS IN TERMS OF DICE RATIOS ON THE 3D BRAIN MRI SEGMENTATION TASK. THE 10-FOLD CROSS-VALIDATION IS ADOPTED TO COMPUTE THE SCORES.

Model	CSF Dice Ratio	GM Dice Ratio	WM Dice Ratio	Avg. Dice Ratio
CC-3D-FCN [29]	0.9250±0.0118	0.9084±0.0056	0.8926±0.0119	0.9087±0.0066
Non-local U-Net [28]	0.9530±0.0074	0.9245±0.0049	0.9102±0.0101	0.9292±0.0050
AEANet	**0.9556±0.0062**	**0.9279±0.0052**	**0.9136±0.0117**	**0.9324±0.0052**

TABLE V

QUANTITATIVE EVALUATION OF THE EQUIVARIANCE TO SPATIAL PERMUTATIONS . SHOWN ARE MEAN ABSOLUTE ERRORS (MAES) BETWEEN OUTPUTS FOR RAW AND PERMUTED INPUT PATCHES.

Methods	Rot. 90	Rot. 180	Rot. 270	Traspose
Self-attention	0.01529	0.02035	0.01518	0.00810
Learned query	**0.01322**	0.01757	0.01369	0.00712
AEA block	0.01351	**0.01689**	**0.01358**	**0.00506**

only if the learned query reduces to constant values at all locations. However, in this case, the learned query is unable to capture common features among the dataset and hence becomes meaningless. The results also indicate the issue related to the invariance property cannot be addressed by data augmentations. In contrast, the AEA block is able to remain spatially permutation equivariance according to the visualization while capturing common features.

On the model level, we quantitatively evaluate the equivariant property by computing the difference, in terms of mean absolute error, between the outputs of raw and permuted input patches. We evaluate models with the self-attention block, the attention with learned query, and the proposed AEA block under rotations and transpose permutations. MAE scores in Table V demonstrate that the proposed AEA block can achieve even better permutation equivariance compared the self-attention block. Note that although the attention with learned query can also achieve a similar level of equivariance, it is unable to learn meaningful query tensors or outputs as the query tensor reduces to constant values over spatial locations.

E. Ablation Studies

We conduct an ablation study to analyze (1) how the shared references and the Batch-aware Attention mechanism help improve the performance of attention-based models and (2) how AEANets benefit from a larger input image size. For a fair comparison, all the models in this subsection are trained with pooled-training.

We first evaluate the performance of AEANets with the following options: Batch-aware Attention excluded, shared references excluded, and shared references with different sizes (16, 32 and 64). The results in terms of ΔPSNR and ΔSSIM are shown in Table VI. Compared to the original attention-based model, GVTNets, applying Batch-aware Attention and shared references renders a performance gain of 0.09 dB and 0.17 dB, respectively. When increasing the size of the shared references, the performance of AEANets also increases.

Regarding the size of input images, we evaluate AEANets on the same testing images but with different input sizes. In particular, we crop each image into patches of a given size,

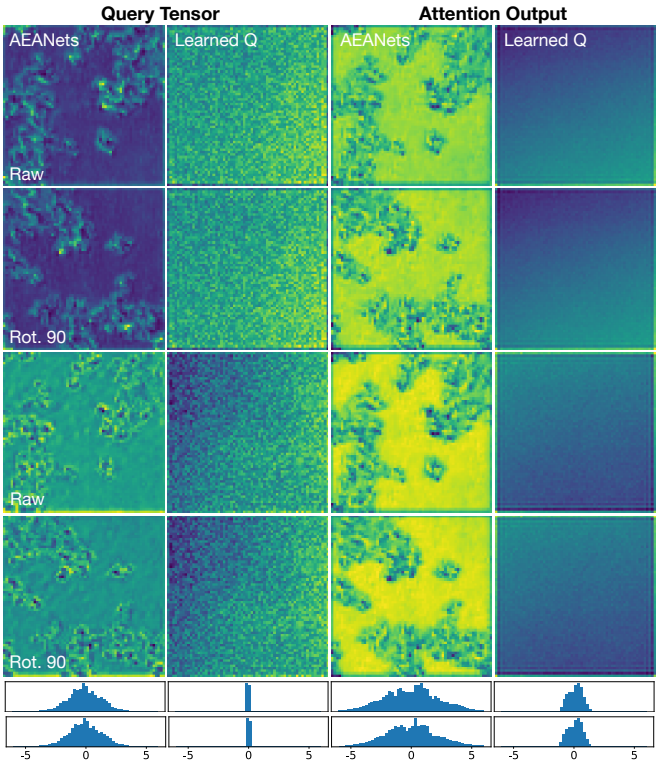

Fig. 6. Visualization of query tensors and attention output tensors of AEANets and Attention with learned queries. Each row corresponds to one channel (out of many) of the tensors. At the bottom are the distributions of values in the tensors with raw and rotated input, respectively.

input the patches to the network and then stitch the predicted patches together as the prediction of the entire image. We evaluate the improvement in PSNR for each size and show the results in Supplementary Figure 9. Among the evaluated alternative, both GVTNet and AEANets benefit from a larger patch size since both are attention-based models. It is interesting to see that the performance of GVTNet with full-sized input can be achieved by AEANets with much smaller input patch sizes. Specifically, the AEANet with shared references of size 64 and the input size 256 reaches the similar performance of the GVTNet with full-sized input.

VII. CONCLUSION

High-quality microscopy images in terms of resolution or noise level are usually desired for better Biomedical and Nanomaterial researches. Computational methods that perform super-resolution and denoising on microscopy images make it possible to obtain high-quality microscopy images more efficiently with lower cost. In this work, we consider the microscopy image-to-image transformation and focus on challenges in the case where both high-quality and low-quality

TABLE VI
PERFORMANCE OF AEANETS WHEN BATCH-AWARE ATTENTION IS EXCLUDED OR THE SIZE OF SHARED REFERENCES ARE DECREASED. THREE BASELINE METHODS ARE ALSO GIVEN FOR COMPARISON.

Methods	ΔPSNR (dB)	ΔSSIM
RCAN	1.59	0.051
U-Net	1.46	0.074
GVTNets	1.87	0.086
Shared Reference (SR) only	2.04	0.085
Batch-Aware (BA) only	1.96	0.084
BA + SR (16)	1.98	0.085
BA + SR (32)	2.02	0.085
BA + SR (64)	**2.10**	**0.087**

images in the training dataset are physically captured. To address the challenges, we have introduced the Augmented Equivariant Attention Networks (AEANets), which is able to utilize shared features among images and inter-image dependencies, and preserve the spatially permutation equivariant property for image-to-image transformation. We have theoretically analysed the property of the proposed attention operator augmented by shared references and the property of existing attention operators as comparisons. And we have conducted experiments to show the effectiveness of AEANets.

REFERENCES

[1] M. Weigert, U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, M. Rocha-Martins, F. Segovia-Miranda, C. Norden, R. Henriques, M. Zerial, M. Solimena, J. Rink, P. Tomancak, L. Royer, F. Jug, and E. W. Myers, "Content-aware image restoration: pushing the limits of fluorescence microscopy," *Nature Methods*, vol. 15, no. 12, pp. 1090–1097, 2018.

[2] Y. Wu, Y. Rivenson, H. Wang, Y. Luo, E. Ben-David, L. A. Bentolila, C. Pritz, and A. Ozcan, "Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning," *Nature methods*, vol. 16, no. 12, pp. 1323–1331, 2019.

[3] E. Nehme, L. E. Weiss, T. Michaeli, and Y. Shechtman, "Deep-storm: super-resolution single-molecule microscopy by deep learning," *Optica*, vol. 5, no. 4, pp. 458–464, 2018.

[4] L. Heinrich, J. A. Bogovic, and S. Saalfeld, "Deep learning for isotropic super-resolution from non-isotropic 3d electron microscopy," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 135–143.

[5] L. Fang, F. Monroe, S. W. Novak, L. Kirk, C. R. Schiavon, B. Y. Seungyoon, T. Zhang, M. Wu, K. Kastner, A. A. Latif *et al.*, "Deep learning-based point-scanning super-resolution imaging," *Nature Methods*, pp. 1–11, 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5998–6008.

[7] Z. Wang, Y. Xie, and S. Ji, "Global voxel transformer networks for augmented microscopy," *Nature Machine Intelligence*, vol. 3, no. 2, pp. 161–171, 2021.

[8] Y. Qian, J. Xu, L. F. Drummy, and Y. Ding, "Effective super-resolution methods for paired electron microscopic images," *IEEE Transactions on Image Processing*, vol. 29, pp. 7317–7330, 2020.

[9] Y. Liu, H. Yuan, Z. Wang, and S. Ji, "Global pixel transformers for virtual staining of microscopy images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 2256–2266, 2020.

[10] H. Yuan, N. Zou, S. Zhang, H. Peng, and S. Ji, "Learning hierarchical and shared features for improving 3d neuron reconstruction," in *Proceedings of the 19th IEEE International Conference on Data Mining*. IEEE, 2019, pp. 806–815.

[11] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[12] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *4th International Conference on Learning Representations*, Y. Bengio and Y. LeCun, Eds., 2016.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[14] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[15] D. Marcos, M. Volpi, and D. Tuia, "Learning rotation invariant convolutional filters for texture classification," in *Proceedings of the 23rd International Conference on Pattern Recognition*, 2016, pp. 2012–2017.

[16] R. Zhang, "Making convolutional networks shift-invariant again," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[17] S. Ji, Y. Xie, and H. Gao, "A mathematical view of attention models in deep learning," Texas A&M University, April 2019. [Online]. Available: http://people.tamu.edu/~sji/classes/attn.pdf

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241.

[19] Z. Wang, N. Zou, D. Shen, and S. Ji, "Non-local U-nets for biomedical image segmentation," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 6315–6322.

[20] T. Zeng, B. Wu, and S. Ji, "DeepEM3D: Approaching human-level performance on 3D anisotropic EM image segmentation," *Bioinformatics*, vol. 33, no. 16, pp. 2555–2562, 2017.

[21] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Proceedings of the 2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2564–2571.

[23] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[24] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 286–301.

[25] 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.

[26] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 2971–2980.

[27] L. Wang, D. Nie, G. Li, E. Puybareau, J. Dolz, Q. Zhang, F. Wang, J. Xia, Z. Wu, J.-W. Chen, K.-H. Thung, T. D. Bui, J. Shin, G. Zeng, G. Zheng, V. S. Fonov, A. Doyle, Y. Xu, P. Moeskops, J. P. W. Pluim, C. Desrosiers, I. B. Ayed, G. Sanroma, O. M. Benkarim, A. Casamitjana, V. Vilaplana, W. Lin, G. Li, and D. Shen, "Benchmark on automatic six-month-old infant brain segmentation algorithms: The iseg-2017 challenge," *IEEE Transactions on Medical Imaging*, vol. 38, no. 9, pp. 2219–2230, 2019.

[28] Z. Wang, N. Zou, D. Shen, and S. Ji, "Non-local u-nets for biomedical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[29] D. Nie, L. Wang, E. Adeli, C. Lao, W. Lin, and D. Shen, "3-d fully convolutional networks for multimodal isointense infant brain image segmentation," *IEEE transactions on cybernetics*, vol. 49, no. 3, pp. 1123–1136, 2018.

[30] A. K. Venkataramanan, C. Wu, A. C. Bovik, I. Katsavounidis, and Z. Shahid, "A hitchhiker's guide to structural similarity," *IEEE Access*, vol. 9, pp. 28 872–28 896, 2021.

[31] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

APPENDIX I
DISCUSSION OF THE INVARIANCE AND EQUIVARIANCE PROPERTIES

We consider the properties under three common spatial permutation cases, *i.e.*, rotation, flipping and shifting.

In a natural image classification task, when the input image is rotated, flipped or shifted, we expect the classification result to remain the same as long as the object to be classified is still in the image, as shown in Figure 1 (left). In this case, the model can benefit from its invariance property and an operator that is spatially permutation invariant can help the model realize such property and hence improve its generalization capability.

On the contrary, when performing the rotation, flipping or shifting on the input of an image-to-image transformation model, we expect the output image of the model to be rotated, flipped or shifted correspondingly, as shown in Figure 1 (right). Hence the equivariance to spatial permutation is desired by the model. In this case, if a spatially permutation invariant operator is included, the equivariance will be violated, since the operator outputs a constant tensor while the input image is rotated, flipped or shifted. Hence, operators with such an invariant property can be inappropriate in image-to-image transformation models and may lead to a performance reduction. It is desirable to use a spatially permutation equivariant operator to preserve the equivariance of the model.

APPENDIX II
PROOF OF THEOREM 1

Proof: When applying a spatial permutation $\mathcal{T}_\pi$ to the input $\boldsymbol{X}$ of a self-attention operator A_s, we have

$$\begin{aligned} A_s(\mathcal{T}_\pi(\boldsymbol{X})) &= \left(\frac{1}{s}\mathcal{T}_\pi(\boldsymbol{Q})\cdot(\mathcal{T}_\pi(\boldsymbol{K}))^T\right)\cdot\mathcal{T}_\pi(\boldsymbol{V}) \\ &= \frac{1}{s}P_\pi\boldsymbol{Q}\boldsymbol{K}^T(P_\pi^TP_\pi)\boldsymbol{V} \\ &= P_\pi\left(\frac{1}{s}\boldsymbol{Q}\boldsymbol{K}^T\right)\boldsymbol{V} \\ &= \mathcal{T}_\pi(A_s(\boldsymbol{X})). \end{aligned} \tag{6}$$

Note that $P_\pi^TP_\pi = I$ since P_π is an orthogonal matrix. Since convolutions with a kernel size of 1 are permutation equivariant, the projected $\boldsymbol{Q} = q(\boldsymbol{X}), \boldsymbol{K} = k(\boldsymbol{X}), \boldsymbol{V} = v(\boldsymbol{X})$ are spatially permutation equivariant with respect to the input $\boldsymbol{X}$. By showing $A_s(\mathcal{T}_\pi(\boldsymbol{X})) = \mathcal{T}_\pi(A_s(\boldsymbol{X}))$ we have shown that A_s is spatial permutation equivariant according to Definition 2.

In comparison, when applying $\mathcal{T}_\pi$ to the input of an attention operator A_Q with a learned query $\boldsymbol{Q}$, which is independent of the input $\boldsymbol{X}$, we have

$$\begin{aligned} A_{\boldsymbol{Q}}(\mathcal{T}_\pi(\boldsymbol{X})) &= \left(\frac{1}{s}\boldsymbol{Q}\cdot(\mathcal{T}_\pi(\boldsymbol{K}))^T\right)\cdot\mathcal{T}_\pi(\boldsymbol{V}) \\ &= \frac{1}{s}\boldsymbol{Q}\boldsymbol{K}^T(P_\pi^TP_\pi)\boldsymbol{V} \\ &= \left(\frac{1}{s}\boldsymbol{Q}\boldsymbol{K}^T\right)\boldsymbol{V} \\ &= A_{\boldsymbol{Q}}(\boldsymbol{X}). \end{aligned} \tag{7}$$

Since $A_{\boldsymbol{Q}}(\mathcal{T}_\pi(\boldsymbol{X})) = A_{\boldsymbol{Q}}(\boldsymbol{X})$, we have shown that $A_{\boldsymbol{Q}}$ is spatial permutation invariant according to Definition 2. ■

APPENDIX III
PROOF OF THEOREM 2

Proof: We let $\tilde{P}_\pi = \begin{pmatrix} I_r & \boldsymbol{0} \\ \boldsymbol{0} & P_\pi \end{pmatrix}$, where P_π is the permutation matrix applied to $\boldsymbol{X}$. Then we have

$$\begin{aligned} A_{\boldsymbol{R}}(\mathcal{T}_\pi(\boldsymbol{X})) &= \left(\frac{1}{r+wh}\mathcal{T}_\pi(\boldsymbol{Q})\cdot\left(\tilde{P}_\pi\tilde{\boldsymbol{K}}\right)^T\right)\cdot\tilde{P}_\pi\tilde{\boldsymbol{V}} \\ &= \frac{1}{r+wh}P_\pi\boldsymbol{Q}\tilde{\boldsymbol{K}}^T(\tilde{P}_\pi^T\tilde{P}_\pi)\tilde{\boldsymbol{V}} \\ &= P_\pi\left(\frac{1}{r+wh}\boldsymbol{Q}\tilde{\boldsymbol{K}}^T\right)\tilde{\boldsymbol{V}} \\ &= \mathcal{T}_\pi\left(A_{\boldsymbol{R}}(\boldsymbol{X})\right). \end{aligned} \tag{8}$$

This shows that the proposed attention operator augmented with shared references is spatially permutation equivariant. ■

APPENDIX IV
NETWORK ARCHITECTURE

The overall architecture of the proposed network is shown in Figure 7.

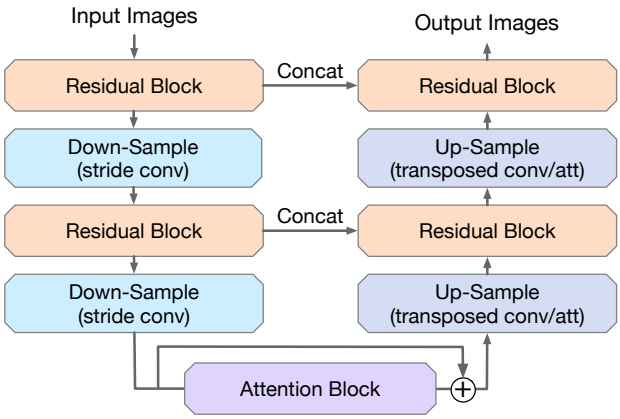

Fig. 7. Network Architecture. We chip our proposed attention block into a U-Net architecture with depth of 3. The skip-connections in the U-Net use concatenation. The down-sampling applies the stride convolution and the up-sampling applies either the transposed convolution or the augmented attention block.

APPENDIX V
IMPLEMENTATION DETAILS AND CONFIGURATIONS

For all three learning tasks on the four datasets, we implement our methods using TensorFlow 1.14 and perform training and testing on a single NVIDIA GeForce RTX 2080 Ti GPU. Below are specific configurations for individual datasets.

For image super-resolution on the Paired 2D EM dataset, we adopt a U-Net with depth of 3 (including 2 down-sampling and 2 up-sampling) as the base structure and replace the bottom block by the proposed AEA block with batch-aware training. The training patches are randomly sampled from training subimages and cropped into a size of 256×256. We adopt a mini-batch size of 8, where 4 samples go through the attention with shared references and the other 4 samples go through the

batch-augmented attention. The two attention operators have their parameters, *i.e.*, projection weights for query, key, and value, shared by setting *reuse=True* at implementation. The model adopts the mean absolute loss as the learning objective and is optimized with Adam optimizer under a base learning rate of 0.0004 with an exponential learning rate decay by 0.5 for every $10,000$ steps. The model is trained for a total of $120,000$ training steps.

For 3D FM image restoration and MR image segmentation, we follow previous works, [7] and [28], respectively, for most model configurations and training settings including the depth of network, type of convolutions, training patch sizes, etc. Models for Planaria restoration and Flywing projection are trained for 80 epochs with batch sizes of 16. As the previous work [7] adopts self-attention blocks at both bottom block and up-sampling blocks, we replace all self-attention blocks by the proposed AEA block for a fair comparison and to better show the effectiveness of the AEA block. For the MR image segmentation, the only difference with the previous work Non-local U-Nets [28] is to replace the attention block by our AEA block. Other network configurations and training settings are kept the same. Models for all ten folds are trained for $300,000$ steps.

According to a recent study [30], the scores computed by different implementations of the SSIM metric may vary. For fair comparisons, we closely follow individual baseline works for the SSIM implementations of each experiment. In particular, for the CARE 3D FM experiments, we use the original evaluation code [1] based on the scikit-image implementation. For the 2D EM experiments, we adopt a matched implementation of Wang et al. [31] (with gaussian weights, sigma=1.5, and covariance sampling disabled) following the previous study [8].

Appendix VI
Effects of Shared References

This subsection provides a discussion on the effects o shared references. Recall that in the learned shared references $\boldsymbol{R} = [\boldsymbol{f}_1, \cdots, \boldsymbol{f}_r]^T \in \mathbb{R}^{r \times c}$ of size r, we call each row vector $\boldsymbol{f}_i \in \mathbb{R}^{1 \times c}$ the feature vector of an abstract pixel distilled from the training images. To illustrate the effects of the shared references, we select three input images and randomly select the feature vectors of four abstract pixels. We visualize how much the Query matrix $\boldsymbol{Q}$ of the three images is correlated to the four abstract pixels. Provided the Query matrix $\boldsymbol{Q}_i \in \mathbb{R}^{wh \times c_1}$ of an input image and the feature vector f_j of an abstract pixel, we visualize

$$\boldsymbol{Q}_i \cdot k^T(f_j) \in \mathbb{R}^{wh \times 1},\ i \in \{1,2,3\},\ j \in \{1,2,3,4\},$$

where $k(\cdot)$ projects the feature vector into the Key space and the dot product indicates the relevance between each pixel in the input image and the learned abstract pixel. We fold $\boldsymbol{Q}_i \cdot k^T(f_j)$ back to the original 2D spatial shape $w \times h$ and visualize it in Figure 8 for each (i, j) in the form of a heatmap. The visualization shows which pixels (or segments) in the input image are tightly related to a given abstract pixel in the shared references.

The four columns on the right show different patterns, suggesting that the abstract pixels in the shared references contain different types of features and are related to different segments of the input images. The abstract pixels in the same column show similar patterns, indicating that features captured by the abstract pixels are shared across images. These two observations tell us that the effect of the shared references matches our expectation.

Appendix VII
Results of Ablation on Training Patch Sizes

Change of performance over different training patch sizes is shown in Figure 9.

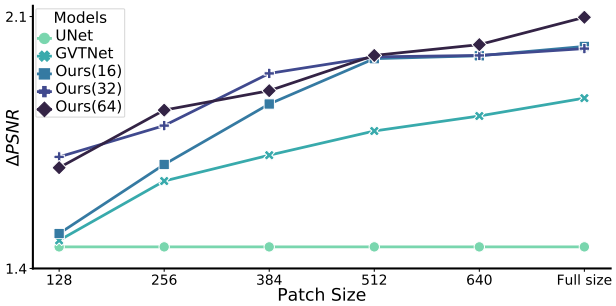

Fig. 9. The output image quality **Δ**PSNR over different input patch sizes. Results are computed on testing images. Numbers inside the brackets in models, *i.e.*, 16, 32 and 64, refer to the sizes of shared references. For all attention-based methods (GVTNet and ours), the output image quality increases when larger input patches are given.

Appendix VIII
Discussions on Limitations and Future Directions

a) Temporal Super-resolution: When capturing a series of microscopy images as a temporal sequence, one has to trade-off between the spatial resolution or quality of each frame and the temporal resolution (fps). Such limitation can be also addressed by extending the advanced image transformation approaches. Besides capturing the sequence in high temporal resolution and computationally obtain high-quality frames, one can also perform temporal super-resolution to directly improve the temporal resolution. The latter case is also an image-to-image transformation problem when considering the temporal dimension as an additional spatial dimension. In both cases, the transformation can benefit from additional information along the temporal dimension, *e.g.*, by aggregating temporal information with attention operators. However, the attention operators requires careful design to enable efficient computation as the temporal dimension brings significantly higher computational cost.

b) Transformation Equivariance with Anisotropic Images: For 3D microscopy, it is common that anisotropic images are captured, where the resolution along the Z-axis (depth) is inconsistent with the other axes. In this case, it is more challenging to achieve spatial transformation equivariance. We do not include experiments related to the anisotropic problem

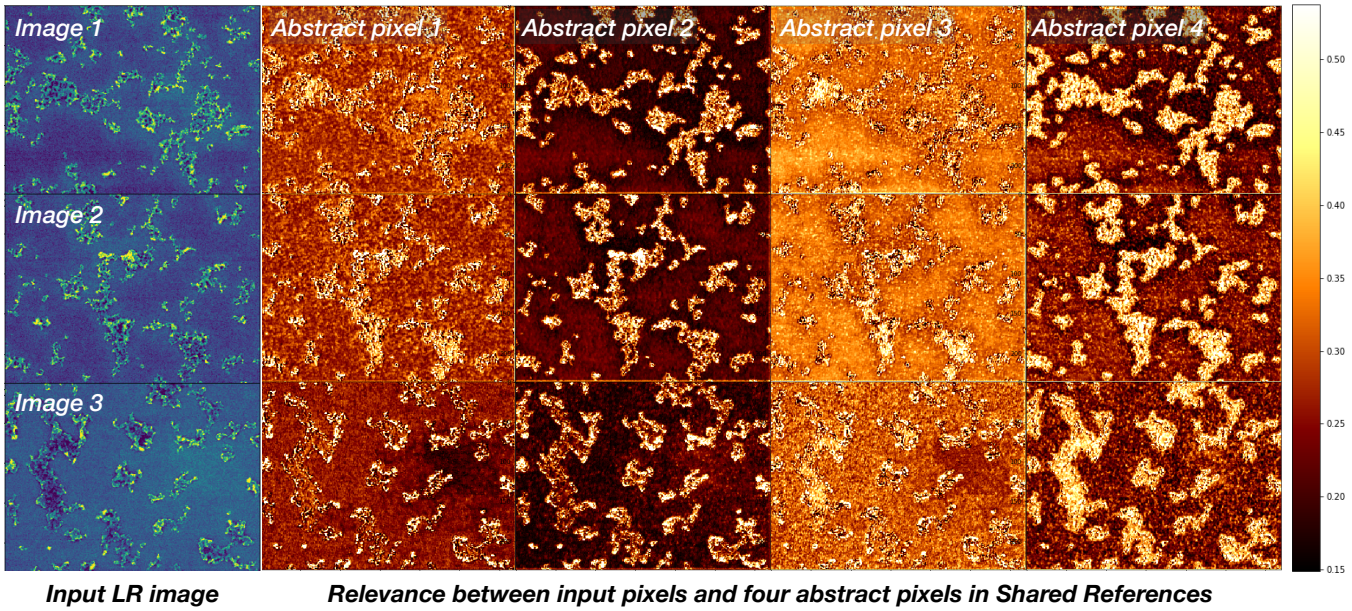

Fig. 8. The visualization of the relevance between pixels in the input image and four randomly selected abstract pixels from shared references on the Paired EM Image dataset. From top to bottom, rows are the visualizations of different input images. From left to right, the first column shows the input images and the rest four columns are the visualizations for the four selected abstract pixels. A higher value in the heatmap indicates stronger relevance.

as it is not part of our claims or conclusions. While our work does not aim at addressing the anisotropic issue of 3D images, the attention-based operators (including the AEA block) can be a promising solution to the anisotropic issue. As the attention-based operators perform non-local aggregation among voxels, a permutation on the input such as transpose that exchanges the resolution of two axes (or spatial distortions) does not change the final output value at corresponding spatial locations. However, the bottleneck of addressing the anisotropic issue lies in the convolutional operators, who rely on preset resolutions along different axes, and performing transpose to the input of such operators will change its outputs. Potential solutions on this issue include cooperating with gating mechanisms to identify inconsistency in resolutions or to adopt fully attention-based networks.