IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 16, 2022, accepted May 30, 2022, date of publication June 2, 2022, date of current version June 7, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3179844*

# Reducing CNN Textural Bias With *k*-Space Artifacts Improves Robustness

**YANIEL CABRERA**[1] **AND AHMED E. FETIT**[1,2]
[1]Department of Computing, Imperial College London, London SW7 2AZ, U.K.
[2]UKRI CDT in Artificial Intelligence for Healthcare, Imperial College London, London SW7 2AZ, U.K.

Corresponding author: Ahmed E. Fetit (afetit@imperial.ac.uk)

The work of Ahmed E. Fetit was supported by the UK Research and Innovation Centre for Doctoral Training in Artificial Intelligence for Healthcare under Grant EP/S023283/1.

**ABSTRACT** Convolutional neural networks (CNNs) have become the *de facto* algorithms of choice for semantic segmentation tasks in biomedical image processing. Yet, models based on CNNs remain susceptible to the domain shift problem, where a mismatch between source and target distributions could lead to a drop in performance. CNNs were recently shown to exhibit a textural bias when processing natural images, and recent studies suggest that this bias also extends to the context of biomedical imaging. In this paper, we focus on Magnetic Resonance Images (MRI) and investigate textural bias in the context of *k*-space artifacts (Gibbs, spike, and wraparound artifacts), which naturally manifest in clinical MRI scans. We show that carefully introducing such artifacts at training time can help reduce textural bias, and consequently lead to CNN models that are more robust to acquisition noise and out-of-distribution inference, including scans from hospitals not seen during training. We also present Gibbs ResUnet; a novel, end-to-end framework that automatically finds an optimal combination of Gibbs *k*-space stylizations and segmentation model weights. We illustrate our findings on multimodal and multi-institutional clinical MRI datasets obtained retrospectively from the Medical Segmentation Decathlon ($n = 750$) and The Cancer Imaging Archive ($n = 243$).

**INDEX TERMS** Texture, bias, artifacts, robustness, MRI, CNNs.

## I. INTRODUCTION

Convolutional neural networks (CNNs) have become the tools of choice for semantic segmentation of biomedical images [1]. However, one notable limitation of modern CNNs is their deterioration under domain shift. When developing a predictive model, we generally assume that the test data belongs to the same distribution seen during training. Yet, such assumption does not usually hold true in realistic settings, such as those observed in clinical workflows. Consequently, when carrying out complex tasks like image segmentation, CNNs tend to underperform in real world scenarios. With magnetic resonance imaging (MRI), there are many factors that can contribute to this, such as hardware differences, variations in image acquisition protocols, acquisition artifacts, as well as mechanical and electronic noise.

We argue that in order to effectively tackle the challenges introduced by domain shift, it is important to re-visit our understanding of CNNs' inner workings. Importantly, there exists a division in our current understanding of how CNNs tend to process images, and there is conflicting evidence in the literature as to how CNNs extract and build meaningful features. On one hand, the 'shape hypothesis' argues that CNNs recognize objects via shape representations, while on the other hand the 'texture hypothesis' postulates a textural-based decision making process [2].

According to the shape hypothesis, CNNs operate through a hierarchical sequence of shape-building steps. This sequence starts with rather simple, shapes-like edges (locally significant changes in contrast) and builds up to more complex visuals such as doors or faces [2], [3]. In a review of biological vision modeling [3], Kriegeskorte explains that CNNs map shape-based visuals into a shape-based and semantic representation. In [4], the authors go at length to describe CNNs as reproducing the human visual cortex, which also works by extracting local features such as edges at low level layers, and global shapes at higher level layers. These views are supported by other empirical studies which have shown that CNNs are especially sensitive to shape features [5].

Diverging from the work above, there is a number of recent studies in the computer vision literature which point to a competing textural bias in deep CNNs [2], [6], [7]. For instance,

The associate editor coordinating the review of this manuscript and approving it for publication was Wenming Cao.

it was shown that CNNs can solve classification problems on ImageNet by simply working with spatially localized features without the need for global shape integration [6]. Further, the experiments conducted in [2] involving stylized versions of ImageNet (using textural transfers) point to CNNs exhibiting a clear textural bias. One of their results suggested that solving the stylized ImageNet problem is harder than working with vanilla ImageNet, since after stylization CNNs could not rely on local textural patches. Studying ways of reducing textural bias as potential remedies for the domain shift problem has therefore made its way into the complex field of biomedical image processing, especially in semantic segmentation of MRI scans [8]–[10]. The recent work of Chai *et al.* [10] used noise injection in the image space as a way of reducing textural bias in deep segmentation CNNs. The authors observed that models trained on images with certain combinations of simple textural filters were able to generalize well to new data with different levels of noise.

We hypothesise that addressing the textural bias phenomenon through the use of realistic MRI acquisition artifacts at training time can improve robustness to: *a)* other types of acquisition artifacts, and *b)* changes in acquisition site, scanner hardware, or imaging protocols. In this paper, we expand on recent findings by studying the effects of textural filters which model realistic noise that manifest during clinical MRI acquisition, through the simulation of *k*-space artifacts. In addition to studying robustness to acquisition noise, we also study the problem of domain shift with respect to the sourcing institutions. Our paper makes three specific contributions:

- We empirically show that CNNs trained on MRI scans can be made robust to a variety of MRI artifacts via stylizations at training time. The focus is on *k*-space corruptions which occur naturally during MRI acquisition.
- We also show that the same artifact stylizations can provide a high level of robustness to out-of-distribution inference for images acquired at hospitals not seen during model training.
- We introduce the Gibbs ResUnet framework, which makes use of a novel layer that can be readily plugged into CNNs to automatically apply Gibbs artifact stylizations. The framework can be trained *end-to-end* to reach optimal stylization for a given segmentation task.

## II. BACKGROUND

### A. TEXTURAL BIAS IN DEEP CONVOLUTIONAL NEURAL NETWORKS

This section provides a review of state-of-the-art work that investigated the CNN textural bias phenomenon on natural images. In 2016, Ballester and Araujo [11] tested pre-trained GoogLeNet and AlexNet on crowd-sourced sketches. Their aim was to investigate what CNNs could learn, as well as their limitations. Their initial hypothesis was that CNNs would be able to match human performance when it came to recognizing objects from the sketches. Instead, the results suggested that in this task, state-of-the-art methods underperformed when compared to human predictions. Although at the time the authors did not explicitly mention it, they were working on a task where textural bias was likely affecting the results.

Gatys *et al.* [12] investigated the area of texture and style transfer, and suggested that the VGG network carries out object recognition tasks without explicitly maintaining a spatial order in the images. Scrambling an image can be seen as a way to texturize it, and the authors illustrated that VGG models predicted the same class label for an image, whether or not the components of the images were scrambled. The authors also performed experiments with style transfer similar to those later performed in depth by Geirhos *et al.* [2]. They applied style transfer on test images and showed that the model classified the images based on the stylization pattern, as opposed to the global shapes within the images.

The 2019 analysis of the more interpretable BagNets architecture by Brendel and Bethge [6] provided further evidence in support of the texture hypothesis. Their work was inspired by earlier non-neural models, namely bag-of-features (BoF), which work by aggregating statistics of local image patches and ignoring spatial relationships of local image features. The authors designed a CNN called BagNet, which was inspired by the BoF model, and studied the receptive fields of the resulting CNNs. Whilst early BoF models performed simple local aggregations, BagNet were able to introduce non-linearity effects into the experiments.

Further evidence comes from studies like [7], where the images were mapped to full textural representations. To generate textures from a given image, the authors used a CNN to extract features of various homogeneous sizes from the image. Using the extracted features, they obtained spatial summary statistics on the features, which corresponded to building a stationary description of the source image. Finally, via gradient descent from a white noise array, the authors created a new image with the obtained stationary description. The main goal of [7] was to introduce a novel approach to textural synthesis via deep neural networks. Yet, an interesting finding of their work was that feeding the CNN textural output into a linear classifier resulted in similar performance to what was achieved by CNN architectures.

Even though the aforementioned studies all suggest that CNNs exhibit a strong textural bias, it was in the work of Geirhos *et al.* [2] where this texture hypothesis was posited. Part of the study involved an expansion of the experiments carried out in [12] changing the texture of tests images to check how the ImageNet-trained CNNs performed. To compare the behavior of the CNNs and humans, the authors carried out psychophysical experiments where human subjects were also asked to classify stylized images. In doing this, they used networks that included GoogLeNet, AlexNet, VGG-16, as well as ResNet-50.

The work of Hermann *et al.* [13] provided a further investigation on the origins of prevalence of CNN textural bias and how model robustness could be improved through specific types of data augmentation at training time. Interestingly, however, Mummadi *et al.* [14] argued that although data augmentation can improve model robustness through stylization, there is no clear correlation between increasing shape bias and improving model robustness.

### B. TEXTURAL BIAS IN THE CONTEXT OF MRI

Findings that support the CNN textural bias argument were also reported in the context of biomedical images. Here we cover relevant examples from the MRI image processing literature, which is the focus of this paper.

A number of studies in the literature involved the use of adversarial training and transfer learning to facilitate domain adaptation, see for example [15], [16]. Shaw *et al.* [17] showed that by simulating artifacts in the data, specifically patient movements in artifact-free MRI data, CNNs can be trained to generalise better and perform more reliably in the presence of artifacts. Hesse *et al.* [8] set out to utilize intensity augmentation to enable domain adaptation for breast MRI segmentation. The domain transfer in question was between T1w and T2w images, with the models trained on T1w data and tested on T2w data. The authors relied on a data augmentation approach, making use of synthetic images and style transfer. By carrying out data augmentation through stylization, the authors tried to probe not just geometric differences that may appear at inference, but also a larger landscape of intensity distributions. They argued that by training on stylized images, the models could focus on global shape features as opposed to local textural and intensity variations.

Whilst the work of Hesse *et al.* [8] focused on removing local textural information, the study by Fetit *et al.* [9] focused on keeping nothing but local textural information. The latter explored constructing segmentation models based on local binary textural maps of the images, without feeding any of the original data to the CNNs. The textural encoding of the images was obtained using the Local Binary Patterns (LBP) algorithm. LBP summarizes local textural patterns for each pixel's neighborhood under the assumption only two measures are needed: the local spatial pattern and gray-scale contrast.

A recent study that investigated textural bias in CNNs is the work of Chai *et al.* [10]. The authors' goal was to use noise (e.g. salt and pepper speckles, Gaussian noise, and median filters) as a way of changing the textural patterns within the training data, in order to improve robustness in segmentation models. Their findings indicate that the models trained on certain noise stylizations generalize to images with different noise composition, suggesting that stylized models can indeed succeed in learning anatomical features and global shape information.

### C. MRI DATA ACQUISITION AND ARTIFACTS

Having discussed state-of-the-art work on CNN textural bias from both natural and magnetic resonance imaging (MRI) perspectives, we now shift the discussion to give a brief overview of how MRI scans are acquired, and how image artifacts normally manifest during acquisition. The literature remains lacking in investigating *k*-space noise that manifests during clinical MRI acquisition in the specific context of CNN robustness, which motivates our work in this paper.

MRI is the name given to tomography obtained via nuclear magnetic resonance (NMR). To obtain the NMR signal, nuclei are exposed to a strong external magnetic field, usually in the range $0.5T$ - $3T$, and a weak radiofrequency (RF) pulse. The strong field forces the nuclei's spins to align parallelly or orthogonally to the field's direction, while the RF pulse induces energy absorption and re-emissions which are recorded at the receivers [18].

The raw data coming from the receivers are amplitudes recorded at fixed frequencies in what is referred to as *k*-space. This *k*-space data is then mapped to an image space *I(x, y, z)* via Fourier transform. In medical settings, various flavours of the NMR signal are used. These different signals are generated by different spin relaxation mechanisms and can be used to create images which highlight tissues with different visual properties. Some of the most common types of MR images are called T1w, T2w, and FLAIR. In a T1w image water and celebrospinal fluid appear dark and fat shows as bright, whereas in a T2w image the celebrospinal fluid appears rather bright. FLAIR images are similar to T2w images but with fluid attenuated to dark and abnormalities show up bright [18]. One refers to the different types of MRI images as 'modalities'.

In terms of artifacts, *Gibbs* artifacts, also called truncation or ringing artifacts in the MRI literature, arise from high contrast regions in the image. These artifacts arise from the Fourier transform process itself, which produces a ringing effect at jump discontinuities, such as in the Heaviside step function. The effects become more apparent in an MRI image when the higher end of the spectrum is not fully resolved; hence the truncation name [19]. In addition to Gibbs, *Spike* artifacts, also called Herringbone or corduroy artifacts, can occur in MRI scans when some of the pixels in *k*-space have abnormal intensities. A third type is the *wraparound* artifact; an aliasing artifact which occurs when some of the recorded data has a phase outside the field of view which is usually set in the range $[-\pi,\pi]$. In this case, the Fourier transform process will generate an image with overlapping spatial coordinates. The aforementioned artifacts were used in this study as a way of introducing realistic noise to the data at training time; the ultimate goal is to develop noise invariant models by introducing the noise into *k*-space as opposed to image space.

## III. MATERIALS AND METHODS

### A. DATASETS USED

In this paper, we worked with two datasets containing clinical MRI scans of brain tumors. The first dataset was obtained from the Medical Segmentation Decathlon (MSD) challenge. It consists of a subset of the 2016 and 2017 Brain Tumor Segmentation (BraTS) challenges, and includes 750 multiparametric MRI scans of patients diagnosed with either glioblastoma or lowergrade glioma [20]. Each sample consists of 4D data, specifically four 3D channels from different modalities: T1w, T1gd, T2w, and FLAIR. The data is multicentric and the acquisition took place during actual clinical practice, spanning various hardware, protocols, and 19 different locations. The targets are three segmentations corresponding to the tumor's sub-regions: edema, enhancing, and non-enhancing tumour.

The second dataset is a joint dataset comprising 243 3D scans from the TCGA-GBM [21] and TCGA-LGG [22] data from [23]. Just like the aforementioned MSD dataset, the TCIA data is pre-operative multimodal and multi-institutional data. It was actually used in the latest version of the BraTS dataset. Importantly to us, this joint dataset contains labels identifying the institutional source for each image. Using this information, we were able to partition the images to fully exclude some institutions from the training set and explicitly study out-of-distribution inference. Neither of the two datasets used in this paper contained time axes, and all the used data was spatial, i.e. $I(x, y, z)$ as opposed to $I(x, y, t)$, where $I$ is the intensity.

### B. DEEP LEARNING FRAMEWORK

Our initial deep learning framework broadly consisted of two parts: *a)* a pre-processing pipeline and *b)* an end-to-end multi-label segmentation model. We utilized MONAI,[1] a PyTorch-based open-source library for deep learning. The benefits of using MONAI include direct access to purpose-built data transforms and different network architectures.

The pre-processing pipeline was structured as follows. Image intensity was standardized by bringing the intensity to zero mean and unit variance; this normalization was carried out channel-wise. Both the image and segmentation maps were then re-sampled to voxel dimensions (2.0, 2.0, 1.5) via a bilinear interpolation on the signal, and nearest-pixel interpolation for the segmentation. The pre-processing pipeline also included the following data augmentation steps: crop each channel in the image and label to 128 × 128 × 64 with a random center using RandSpatialCropd; randomly flip about the 0-axis with probability 0.5 using RandFlipd; scale the image intensity by 10% and probability 0.5 using RandScaleIntensityd; and shift intensity with offset 0.1 and probability 0.5. The final step in the pre-processing pipeline was to introduce the *k*-space MRI artifacts into the data via one of the designed stylization transforms, depending on the experiment. Subsection III (C) provides a summary of the simulated artifacts used in this paper.

Each segmentation model was trained using the Residual U-Net (ResUnet) developed in [24]. The choice of architecture was based on the wide success of the conventional U-Net architecture of [25]–[27] in biomedical segmentation tasks. For the additional benefits of stability and speed, we decided to use the ResUnet network since residual learning is an important strategy for deep networks. The encoder of the network uses 5 stacked residual blocks of depth 2 (12 total convolutional layers) to map the data into a 256-dimensional feature space.

For the loss, we used the Dice loss function. The Dice similarity coefficient (DSC) was used an evaluation metric [28]. A sigmoid activation function was applied within the loss function implementation. This activation layer is commonly moved from the network into the loss function to mitigate the vanishing gradient problem.The following hyperparameters were used: Number of epochs = 110; Optimizer = AMSGrad Adam; Learning rate = $10^{-4}$; Weight decay = $10^{-5}$. These were the hyperparameters used for all the experiments, although one exception is that the number of epochs was larger when 4D data was used (180 as opposed to 110).

### C. SIMULATED ARTIFACTS IN K-SPACE

The *k*-space data represents the spatial frequency information of the scan and it is related to the image data via the 3D Fourier transform. Therefore, in order to implement the artifacts discussed below, the first step was to generate *k*-space data from their corresponding image data by carrying out a Fourier transform.

#### 1) GIBBS

To simulate Gibbs artifacts, we implemented a filter which truncates the frequency content of the scan by masking out the outer part the *k*-space domain of the data. With this filter, one can specify the strength of the transform by how much of the *k*-space volume is kept. The application of this filter at different intensities used in the experiments is shown in Figure 1. The strength of the artifact is parametrized by a floating number $r$. When $r$ is equal to the largest diagonal of the image, one obtains the identity transform, and as one shrinks $r$ the ringing effects appear; at the other extreme of $r = 0$ the image would be mapped to a constant zero.

#### 2) SPIKE

We designed a filter which inserts an anomalous value at a given index location in the *k*-space domain of the data. In the experiments, the anomalous value (intensity) was fixed, and its location was allowed to be sampled uniformly from all the 3D locations in the image as shown in Figure 2. As one can see in the figure, dialing up the intensity obscures the underlying brain structure since the plane wave amplitudes dominate at most pixels.

#### 3) WRAPAROUND

This artifact was simulated by obscuring every other sample in each of the *k*-space dimensions. This is done by multiplying by a mask that is a 3D grid, where each pixel takes values in $\{q, 1\}$, with $q \in [0, 1]$. By choosing the intensity of the mask (the value of $q$), one can parametrize how much the wraparound shows up in the image. Figure 3 contains examples of this artifact at various strengths. The wraparound happens along the three spatial dimensions.

[1] https://monai.io/

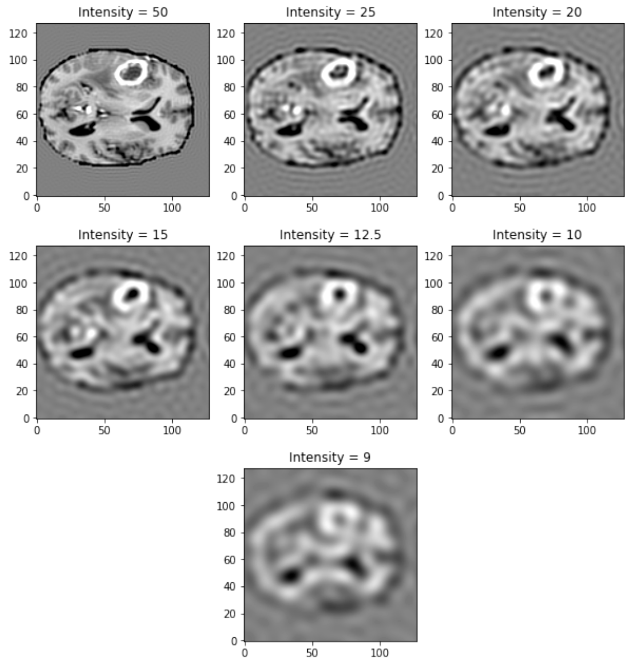

**FIGURE 1.** Applications of the Gibbs artifact filter at various intensities on an axial brain MRI slice. Larger $r$ values result in lower artifact severity.

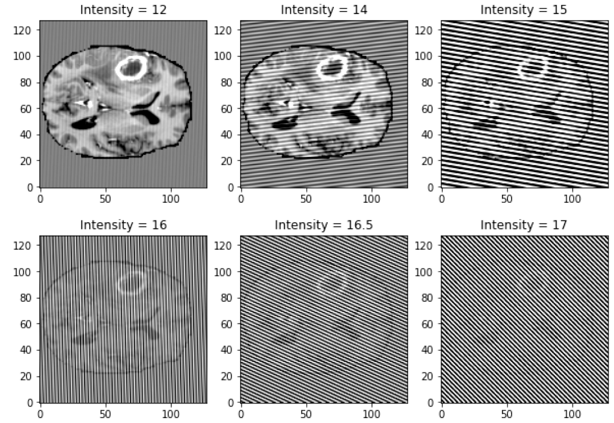

**FIGURE 2.** Applications of the spike artifact filter at various intensities on an axial brain MRI slice. The location of the anomalous $k$-space pixel is allowed to randomly vary hence the 3D plane waves have arbitrary direction in each case.

### D. SIMULATED NOISE IN THE IMAGE SPACE

In addition to introducing the $k$-space artifacts mentioned above, we also simulated salt-and-pepper noise directly in the image space. This fourth stylization was implemented for comparison against the approach discussed by Chai *et al.* [10]. The transform is parametrized by a factor $p \in [0, 1]$ and works pixel-wise by assigning each pixel a value $x$ drawn from $[0, 1]$ uniformly. At each pixel if $x > p$, the amplitude is left untouched. Else, if $x \in [0, p/2]$ the amplitude is set close to white; otherwise if $x \in [p/2, p]$ the amplitude is set close to black. Examples of images with the salt-and-pepper transform applied are shown in Figure 4. The amplitude of the affected pixels was set to either half the maximum or minimum amplitude of the image. This choice was made to keep a constant relative contrast between the affected and unaffected pixels for each image.

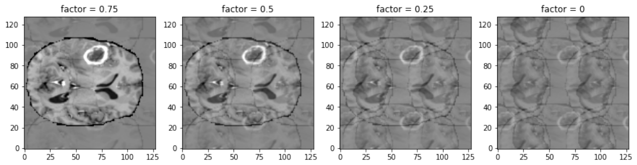

**FIGURE 3.** Examples of wraparound artifact simulated at different severity levels. Factor = 1 corresponds to identity mapping; factor = 0 means that the wraparound artifact shows up with the same amplitude as the original data.

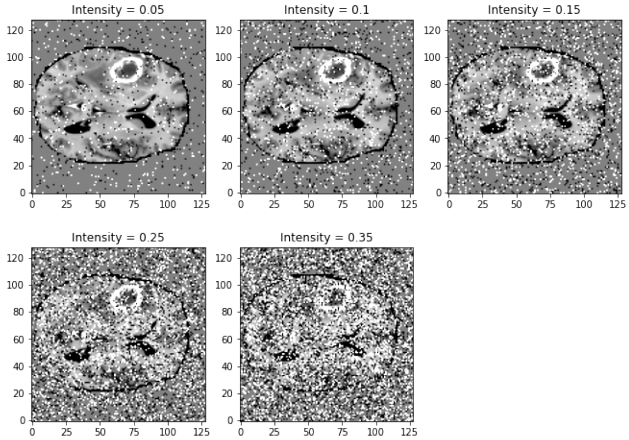

**FIGURE 4.** Examples of salt-and-pepper noise simulated at different levels of severity. Factor = 0 corresponds to the identity mapping; factor = 1 is pure uniform noise.

### E. EXPERIMENTS ON ROBUSTNESS TO ARTIFACTS

As seen in the literature on natural images of [2], [12] and on MRI data [8]–[10] CNNs can express a marked bias towards the textural attributes of an image. Particularly interesting are the results of Chai *et al.* [10], where the authors showed that simple textural changes applied to the image by injecting noise can improve robustness to distributional shift. Differently from the other works on textural bias, the authors in [10] did not apply full stylizations which completely corrupted the image (e.g. via scrambling or style transfer).

In this paper, our first set of experiments aimed to expand on the observation that careful use of noise at training time can be used to improve model robustness. Whilst Chai *et al.* discussed noise that can appear during post-processing (e.g. blurring with Gaussian filters and impulse noise modeled with salt-and-pepper), we here experiment with artifacts that are directly related to MRI data acquisition, namely Gibbs ringing, spike artifacts, and wraparound aliasing. There are two reasons why employing such artifacts is important:

- MRI acquisition artifacts can strongly contribute to institutional domain shift since different imaging hardware or protocols can result in different rates of occurrence of these artifacts.
- Since MRI data acquisition happens in $k$-space before the data is transformed to the image space, artifacts originating in $k$-space have a more 'intrinsic' signature

on the images, making them highly pervasive. Their effects may therefore strongly influence model training down the line.

We also made use of salt-and-pepper in this study as a non *k*-space artifact, and as a way to link the work of Chai *et al.* with the experiments of this project.

For each artifact type, we trained a model whose training data has a fixed intensity of the artifact applied. By intensity we refer to the aggressiveness or strength of the artifact. For each artifact, we trained models with various (fixed) intensities as well; each model was then tested on datasets corresponding to other models. The expectation is that there is a trade-off between possible robustness and performance degradation coming caused by the noise. This trade-off appeared in the literature in different forms, e.g. in [6] with the $q$ in BagNet-q, and [9] with the radius of the LBP algorithm.

In addition to testing models on different intensities of the same artifacts, we also experimented with cross-artifact inference. For example, each model trained with a Gibbs artifact was then tested on wraparound test datasets. The goal was to see whether the artifacts stylizations can help generalize the models to other types of noise.

This set of experiments was carried out using the Decathlon dataset described above. Since the dataset presents a multilabel segmentation task with three different labels, we compared the models using the mean Dice score averaged over the three outputs. Each stylization was applied as the last transform in the pre-processing pipeline before the data was passed to the segmentation network.

For training, we used 388 samples out of the 484 available in the Decathlon data set. The remaining 96 samples were partitioned into validation and testing sets of 48 samples each. All models based on the Decathlon dataset were trained for 180 epochs.

### F. EXPERIMENTS ON ROBUSTNESS TO HOSPITAL DISTRIBUTION SHIFT

A strong check of the effectiveness of *k*-space artifacts in tackling domain shift is to test whether there is an improvement in out-of-distribution robustness when the stylized models are used. For this set of experiments, we employed the same pre-processing pipeline described in the previous section. We also used the same overall convolutional ResUnet architecture, albeit modified to accept and output a three dimensional tensor since the TCIA data is 3D, as opposed to 4D. Given the lower dimensionality of the data, the models were trained for 110 epochs as opposed to 180 in the previous experiments.

The models were trained in the presence of stylizations, while keeping a held-out set where all images come from one institution that was fully excluded from the training set. A second held-out test set was also built by sampling data points from the same institutions used in the training set to create the in-distribution test set. All models, including the baseline, were trained using the same scheme and same number of epochs. Each model was the then tested on the in-distribution test set as well as the out-of-distribution test set.

In terms of evaluation, there were two key measures: the performance of the models on the in-distribution test set, and how the performance *changes* when tested on the out-of-distribution set, i.e. the level of robustness.

### G. THE LEARNABLE GIBBS LAYER

In the experiments described above, the stylization parametrization (e.g. the *k*-spike intensity) was fixed for each model. Here, we take a different approach by replacing the stylization pre-processing step with a learnable layer within the segmentation network. With this change, we can allow the training process to find an optimal combination of the stylization and the segmentation architecture. The new ‘stylized’ architecture consists of a Gibbs layer followed by the ResUnet. Inputs and outputs are still 3D tensors. Throughout the paper, we will refer to our stylized network as *Gibbs ResUnet*.

The hyperparameters controlling the intensity of the stylization filters do not make it as a variable into the loss function, even when they are attached to the computational graph of the architecture. In the case of the Gibbs layer there is a hyperparameter, which we call $\alpha$, controlling the size of a *k*-space mask. Hence, this hyperparameter is not applied directly to the data as a weight. For this reason, we implemented the optimization of the Gibbs layer via a finite-difference implementation of minibatch SGD. Instead of using automatic differentiation, at each step the algorithm updates the following for each data minibatch:

$$\alpha \to \alpha - \eta \frac{L(\alpha + h) - L(\alpha)}{h} \tag{1}$$

where $h$ is the finite-difference step, and $\eta$ is the learning rate. The values of $h = 0.01$ and $\eta = 0.02$ were empirically found to work well. Note that the above equation is schematic; the parameter $\alpha$ goes into the the loss $L$ via the model. To train the whole model, updates were done in two steps within each iteration: *(1)* updating the ResUnet using the Adam optimizer, and *(2)* updating the Gibbs layer using the finite-difference gradient descent.

## IV. RESULTS AND DISCUSSION

In this section, we present and discuss the results of the robustness experiments described above. It is important to highlight that by robustness we refer to the difference between a model’s best and worst performances over different settings. In other words, a robust model would have little variation under different cases of domain shift presented to it, while a non-robust model would show a drop of performance when presented with domain shift. Of course, a model that underperforms across the board is not a useful model, albeit arguably “robust” if it can maintain the same level of performance throughout the experiments.

### A. ROBUSTNESS TO ARTIFACTS

Below, we present the results of: *(1)* using the same artifacts during both training and testing, as well as *(2)* using artifacts different from those introduced during training.

#### 1) USING THE SAME ARTIFACT TYPE DURING BOTH TRAINING AND TESTING

In most cases, there was a general trade-off between robustness and performance on the baseline data. When restricting inference to within each stylization type, the stylized models were found to be more robust to varying degrees of the noise than the baseline model. Specific results are detailed below.

##### a: GIBBS-STYLIZED MODELS

Dice scores for Gibbs-stylized models on Gibbs test data are shown in Figure 5(B). The right-most $x$-axis value of 50 corresponds to a very mild application of the artifact, which leaves the data similar to the original version (see Figure 1 for visualizations of the tested filter strengths). The solid black curve represents the non-stylized baseline model performance. Over the range of tested Gibbs strengths, the baseline model dropped in performance by roughly 18%. Meanwhile, the stylized models showed much lower variance across the $x$-axis. For example, the performance of the Gibbs-12.5 model (model trained on Gibbs 12.5 data) varied by only 4%; and that of the Gibbs-9 model varied by only 1.4%, suggesting a high level of robustness. At the same time, there was a degradation of the models' performance on the baseline data as we increased the Gibbs factor.

##### b: SPIKES-STYLIZED MODELS

Observations similar to the Gibbs case hold for the models trained and tested on the spike artifacts data. Performance for these models is shown in Figure 6(C). Here, one sees a very marked drop in performance of the baseline model as it probes increasing intensities of the artifact (moving to the right along the $x$-axis). The baseline model Dice score varied by as much as 90%. On the side of the artifact intensity spectrum, the spikes-16 model score varied by only 4% while performing close to the baseline model on the baseline data, which suggests a high level of robustness. The other models' performances interpolate those of the baseline and spikes-16 models. Examples of images belonging to each tested intensity of the spike artifact are in Figure 2.

##### c: WRAPAROUND-STYLIZED MODELS

Figure 7(A) shows the performance of the wraparound models on the different wraparound test sets. These models demonstrated robustness to varying intensities of the artifact. Only the edge case (the wrap0 model) underperformed the baseline model. Noteworthy, in this edge case the aliasing noise was as strong as the signal itself. Another difficulty is that the artifact itself was drawn from the same distribution as the underlying image. That is, the artifact consisted of multiple superpositions of the image along each axis (see Figure 3). It is therefore possible that those characteristics of the artifact make it difficult for the network to choose the right signal to focus on.

##### d: SALT-AND-PEPPER STYLIZED MODELS

The last case was the salt-and-pepper stylized models. Their performance across the salt-and-pepper test sets is shown in Figure 8(D). The robustness of the salt-and-pepper models to unseen levels of noise was remarkable. While the baseline model showed a drop of 40% across the datasets, some of the stylized models generally stayed consistent in performance. The cost of underperforming on the baseline test set was only about a 3% drop in performance for the S&P-25 model.

#### 2) USING DIFFERENT ARTIFACT TYPES DURING TRAINING AND TESTING

We here present comparisons of the models trained on data stylized with one artifact type, but tested on a different stylization category. Generally, stylized models did not always beat the baseline model when tested on differently stylized datasets. Yet, in some cases, e.g. the spikes-12 model, robustness was demonstrated across all the stylizations considered in this study. Details of our findings are shown below.

##### a: SPIKES-STYLIZED MODELS

The performance of models trained on spikes-stylized data strongly outperformed the baseline model when tested on S&P-stylized data, as shown in Figure 8(C). This is an interesting result since the image and $k$-space characteristics of the spikes artifact and the salt-and-pepper noise are very different. In Figure 5(C), we see that these models also perform better than the baseline on Gibbs-stylized data. On the wraparound data, the spikes-12 model was the only one to slightly outperform the baseline model. These observations, together with the performance of the spikes-12, suggest that the spikes-12 model is more robust than the baseline model against all noise types and intensities considered here.

##### b: GIBBS-STYLIZED MODELS

Models trained on Gibbs data performed as well as, or worse than, the baseline model on the spikes test data, wraparound test data, and S&P test data as shown in the plots (B) of Figures 6, 7, and 8 respectively. For mild applications of the filter at training time, the stylized models showed a very close performance to the baseline model on the cross-artifact tests.

##### c: WRAPAROUND-STYLIZED MODELS

The wraparound-stylized models mostly underperformed the baseline model on differently stylized data. The performance curves of the model with the mildest wraparound factor (wrap-75) closely mimicked those of the baseline model as seen in Figures 5(A), 8(A), and 6(A). Since the wrap-75 model also outperformed the baseline model on wraparound test sets, we see that it does not a price on the differently stylized tests for such performance boost.

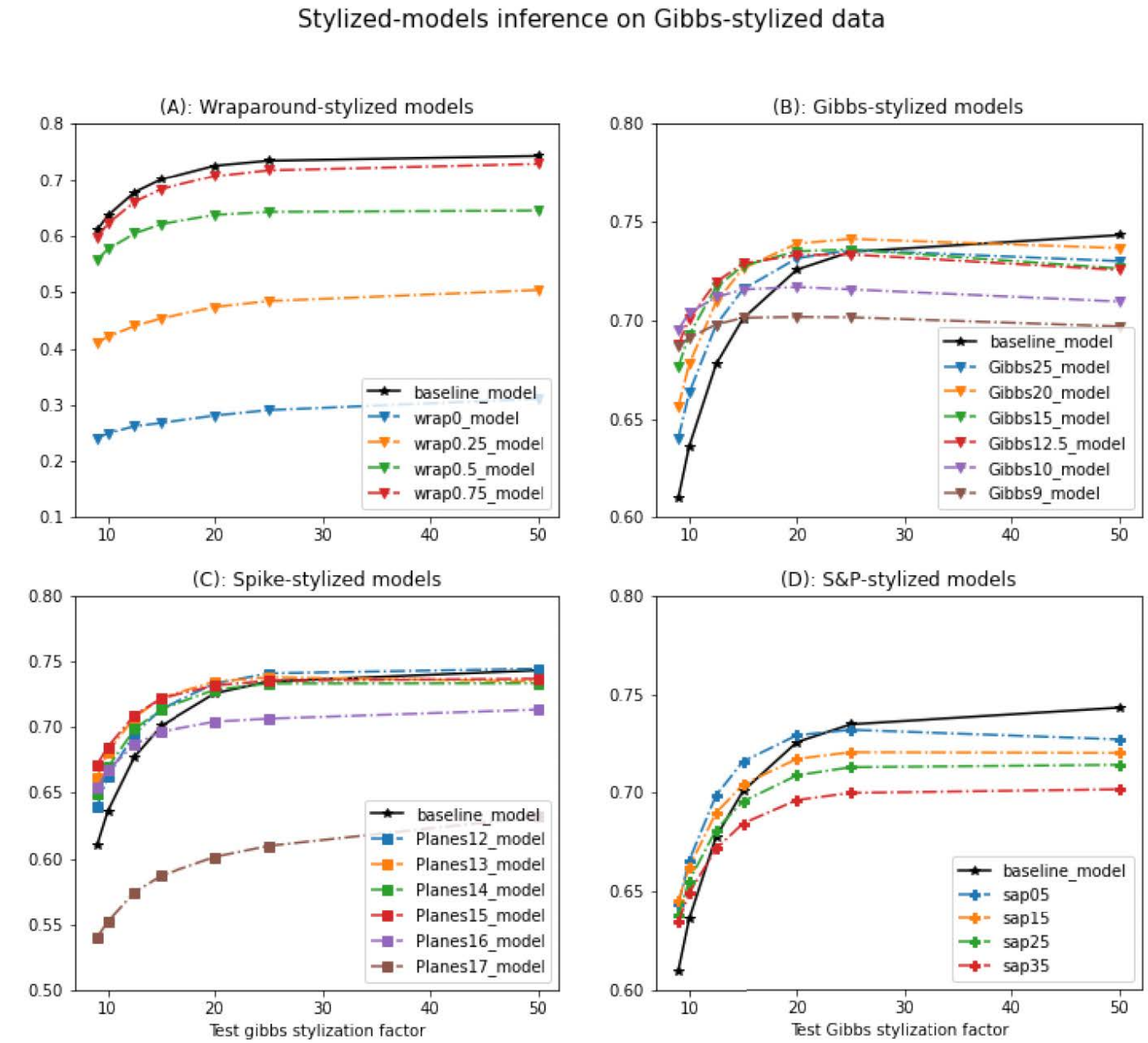

**FIGURE 5.** Inference of stylized models on the Gibbs stylized test sets. The x-axis labels the test sets stylized at different intensities of the Gibbs artifact. The intensity of the artifact decreases with increasing factor.

### *d: S&P-STYLIZED MODELS*

The S&P models showed a mixed performance on differently stylized test sets. Inference on spike data (Figure 6(D)) indicates that these models all performed better than the baseline model, while paying a small price on the baseline test set. Yet, switching to the wraparound test sets, the S&P models performed about as good as the baseline (Figure 7(D)); and on the Gibbs test sets their gains were mostly offset by the price paid on the baseline test set (Figure 5(D)).

## *B. ROBUSTNESS TO HOSPITAL DOMAIN SHIFT*

The results presented so far indicate that in some cases the stylized models can exhibit good levels of robustness on artifact intensities and categories not seen during training. In medical settings, a scenario of relevance is when the segmentation model is presented with MRI images coming from a new hospital which was not in the set of institutions seen during training. Here we present results which show that *k*-space artifact stylizations can be used to achieve robustness on out-of-hospital-distribution images. To check that each model has been trained sufficiently during the selected number of epochs (number of epochs = 110), we plot the learning curves for each case in Figure 10. The plots show that all models have learned on par with the baseline model even if they are handling stylized datasets.

The results of the Gibbs-stylized models are displayed in Figure 9. It is important to emphasize that neither of the test sets were stylized. The $x$-axis represents the intensity of the artifact, while the $y$-axis on the left represents the mean Dice score. We represent the gap between the in- and

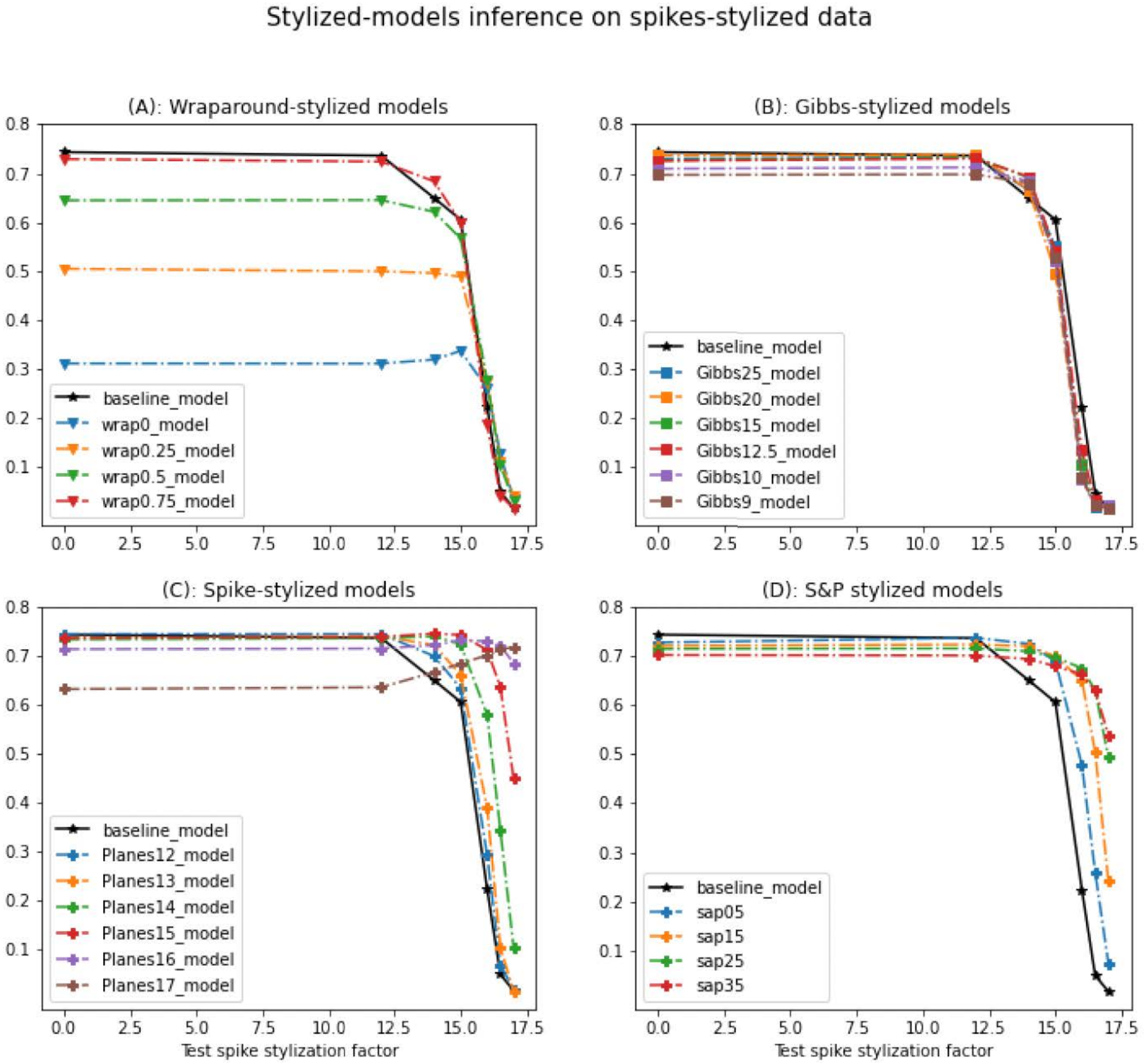

**FIGURE 6.** Inference of stylized models on the spikes stylized test sets. The *x*-axis labels the test sets stylized at different intensities of the spikes artifact. The intensity of the artifact increases with increasing factor.

out-of-distribution scores by a bar, where the higher and endpoints of the bar mark each Dice score (higher is always in-distribution). For ease of comparison against the un-stylized baseline model, we plotted a light blue band, which is the baseline performance gap. To get a clear measure of the decrease or increase in the difference between the performance of each model in- and out-of-distribution, we plotted the $\Delta$-ratio, which we defined as:

$$\Delta D/\Delta D_0 \tag{2}$$

where $\Delta D$ is the difference in Dice scores between in- and out-of-distribution test data; $\Delta D_0$ is this difference computed for the baseline model. For example, a value of 1 for this $\Delta$-ratio means that the given model has a performance gap equal to that shown by the baseline model.

As per Figure 9, we can observe that for some mild applications of the Gibbs artifact (mild being a value greater than 35), we can obtain a better out-of-distribution performance. Of the stylized models, two versions exhibited this characteristic: Gibbs-45 and Gibbs-65. For both of these models, the out-of-distribution score was higher than that for the baseline (they are above the lower bound of the blue band). At the same time, their normalized Dice difference was close to or less than 1. Other stylized models also showed $\Delta$-ratio values below 1, meaning that their out-of-distribution performance was closer to the in-distribution one than in the case of the baseline model. Nevertheless, those models also underperformed the baseline model.

The results of the spikes-stylized models are displayed in Figure 11. Similar to the Gibbs-stylized model, we see that

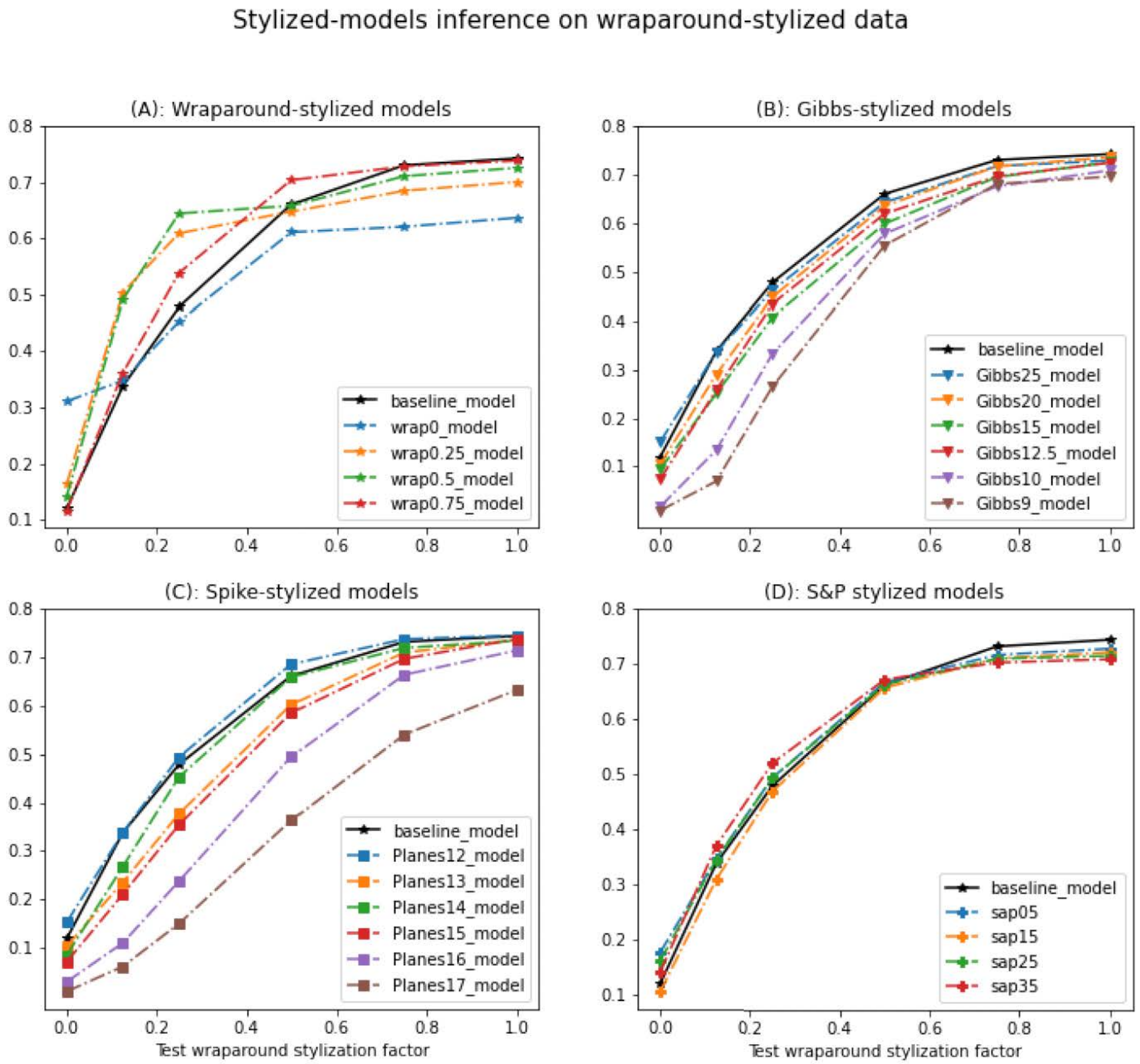

**FIGURE 7.** Inference of stylized models on the wraparound stylized test sets. The $x$-axis labels the test sets stylized at different intensities of the wraparound artifact. The intensity of the artifact decreases with increasing factor.

the spikes artifacts could also result in models which perform better than the baseline when tested on the out-of-distribution set. For those cases ( i.e. spikes-9 and spikes-10), we also see that the Δ-ratio is close to 1. The learning curves for the spikes-stylized models are shown in Figure 12.

### C. THE LEARNABLE GIBBS LAYER

We first carried out a reality check, where we trained the Gibbs layer with a fixed, pre-trained ResUnet (Figure 13). The resulting curve is plotted for two separate runs with different Gibbs layer initializations. Indeed, the expected behavior is observed with $\alpha$ gravitating towards $\alpha = 1$, which is the extreme case where the Gibbs layer coincides with the identity mapping. This behavior indicates the the Gibbs layer implementation (or rather, the minibatch Stochastic Gradient Descent on it) was able to search the solution landscape for optimal solutions.

With regards to the end-to-end, Gibbs ResUnet models trained with the novel Gibbs layers, those were able to outperform the baseline model on the out-of-distribution set, but the starting state of the Gibbs layer had a clear effect on the final state. To check the stability of the learning process, we trained various model versions, each with a different given starting parametrization of the Gibbs layer (if not specified, the parameter $\alpha$ would be sampled uniformly upon initialization). In all cases, the overall Dice loss of the model followed a similar decay as the baseline model. This behavior is illustrated in Figure 14. The corresponding trajectories for the Gibbs parameter $\alpha$ are shown in Figure 15.

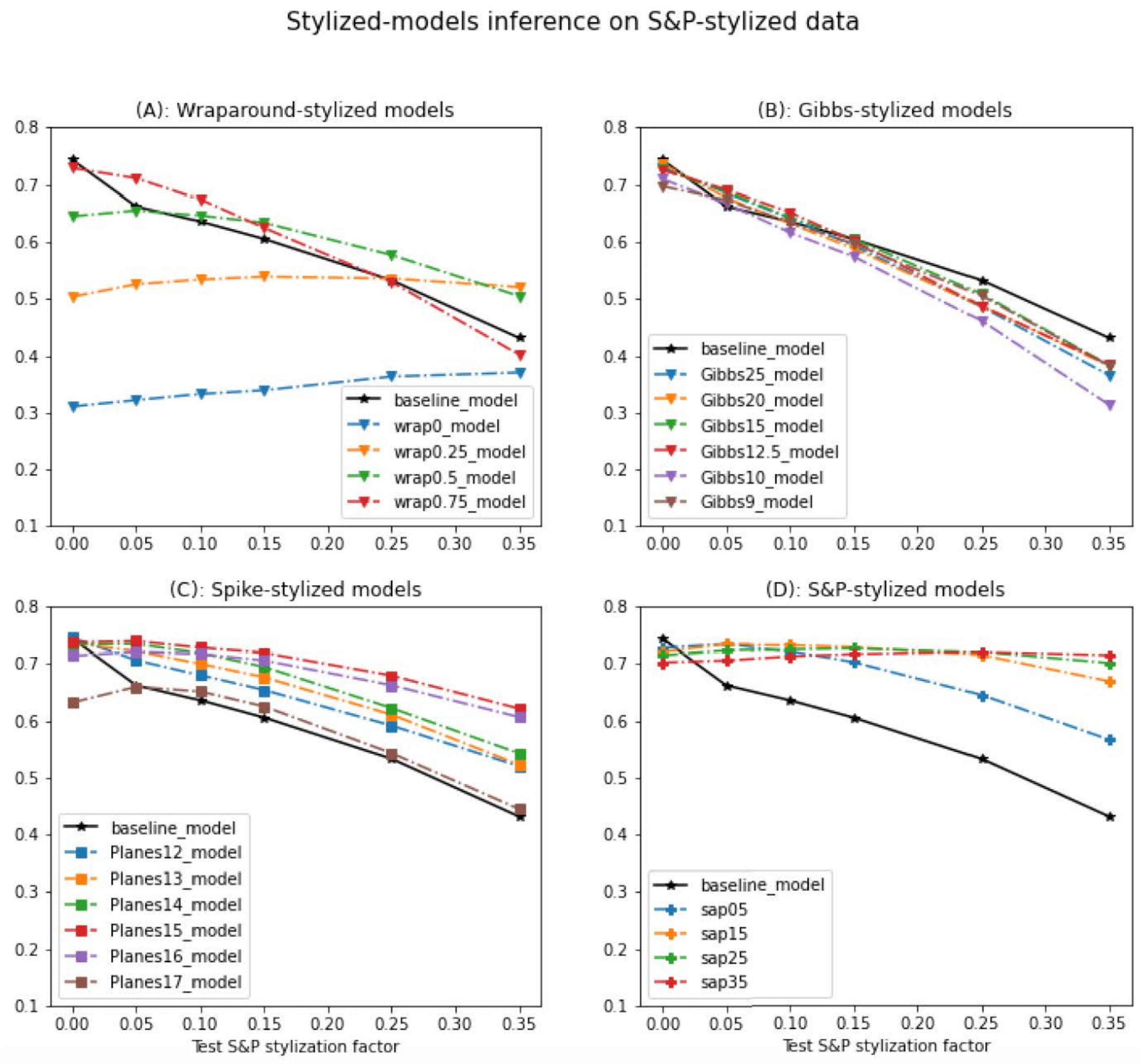

**FIGURE 8.** Inference of stylized models on the salt-and-pepper stylized test datasets. The x-axis labels the test sets stylized at different intensities of the salt-and-pepper artifact. The intensity of the artifact increases with increasing factor.

As per Figure 15, we can see that the finite-difference SGD was able to approach an optimal state for each model. Yet, the learned optima were not unique and clearly depended on the initial parametrization $\alpha_0$. The initialization dependence was apparent since for $\alpha_0 > 0.8$ the final value was around 0.8, while the rest of the curves settled at around 0.5 - 0.6. Such behavior suggests the presence of non-unique solutions, or the possibility that the learning algorithm was simply not able to reach the true optima in some cases.

In Figure 16, we summarized the inference results for each of the Gibbs ResUnet models. Again, for each model, there is a segment whose upper and lower endpoints mark the in- and out-of-distribution mean Dice scores. The models showed better robustness on out-of-distribution data than the baseline. Most of the models, however, paid the price of underperforming on **in**-distribution data, except for the model with starting parameter $\alpha_0 = 0.75$. A visual example comparing the predictions of the baseline and the $\alpha_0 = 0.7$ stylized models on an in-distribution test image is show in Figure 17; a similar example using an out-of-distribution image is shown in Figure 18.

## V. OPEN-SOURCE SOFTWARE CONTRIBUTION

For reproducibility, we shared our implementation of stylization filters on MONAI, an open-source library that is part of PyTorch ecosystem. We designed the implementations to work with minimal user input and with default parameters that allow the code to work out-of-the-box. For each filter, we designed two versions: one which applies the artifact directly, and another one which applies it randomly. The

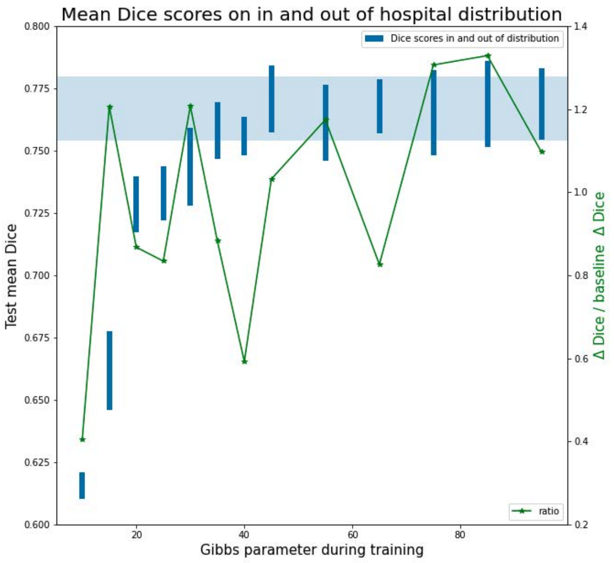

**FIGURE 9.** Performance of Gibbs-stylized models on in- and out-of-distribution data. Each vertical bar represents the change in mean Dice score when a model was tested on in- and out-of-hospital test sets. For comparison, the light blue band represents this difference for the un-stylized baseline model.

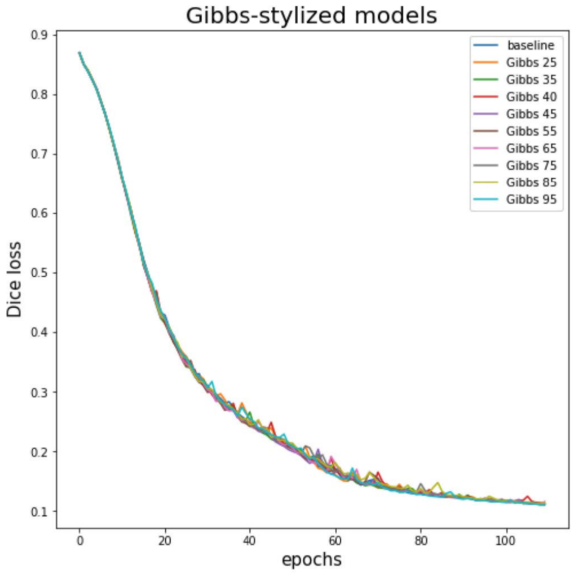

**FIGURE 10.** Learning curves for Gibbs-stylized models.

second version can be used as a pre-processing data augmentation tool by the community to improve the robustness of CNNs trained on MRI data.

All the implementations can readily work with 2D and 3D data. They were also designed to work with Numpy arrays and PyTorch tensors. The user does not need to specify the shape or data type on instantiation, as the code will determine that information automatically when it is called.

To ensure reliability, exceptions were placed on user-set hyperparameters to avoid any unintended results. Any functionality that is common to the various transforms has been relegated to super classes for readability and ease of code maintenance. Lastly, all the transforms were written in with

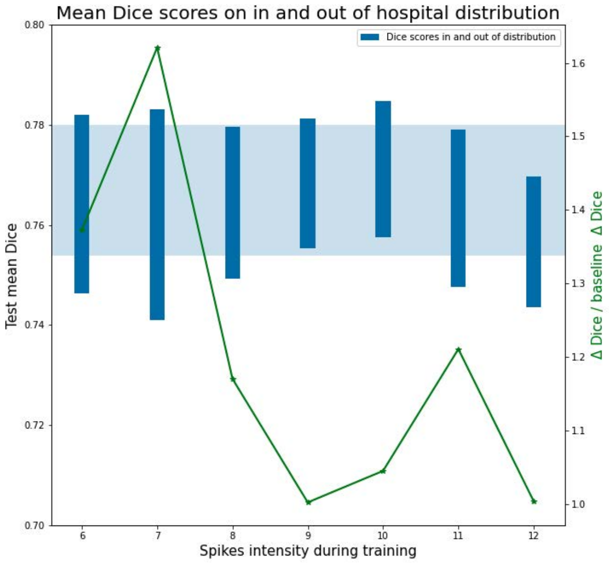

**FIGURE 11.** Performance of spikes-stylized models on in- and out-of-distribution data. Each vertical bar represents the change in mean Dice score when a model was tested on in- and out-of-hospital test sets. For comparison, the light blue band represents this change for the un-stylized baseline model.

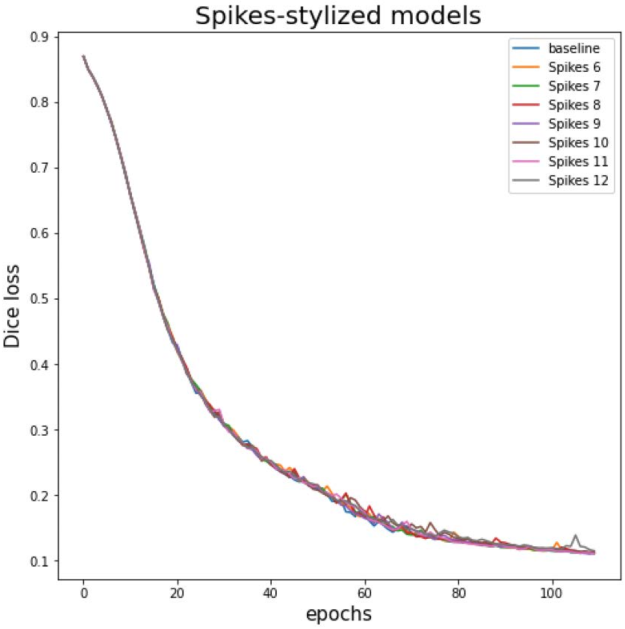

**FIGURE 12.** Learning curves for spikes-stylized models.

multi-threaded safety in mind. Thus, the transforms avoid mutating their own states and can run on multi-processing architectures.

The open-source implementations are listed below, and their full documentation can be accessed through MONAI's documentation pages.[2]

- **monai.transforms.KSpaceSpikeNoise** This is a deterministic transform which applies the *k*-space spike artifact on input images.

[2] https://docs.monai.io/en/latest/transforms.html

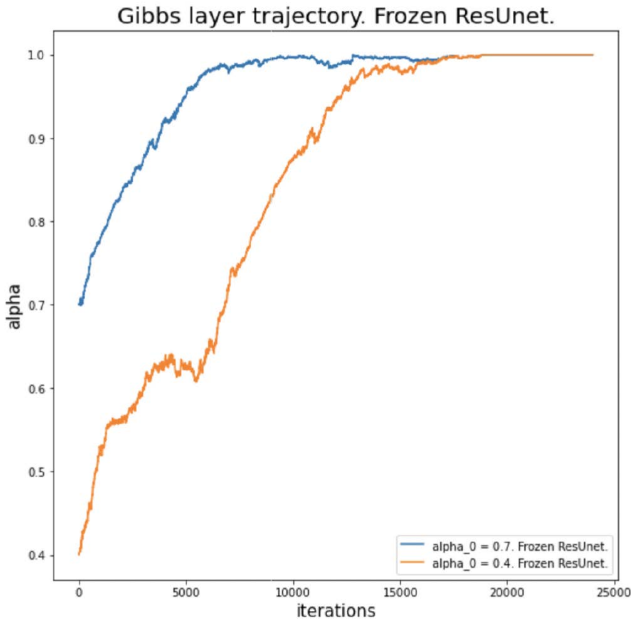

**FIGURE 13.** Gibbs layer trajectories when used in conjunction with a frozen ResUnet.

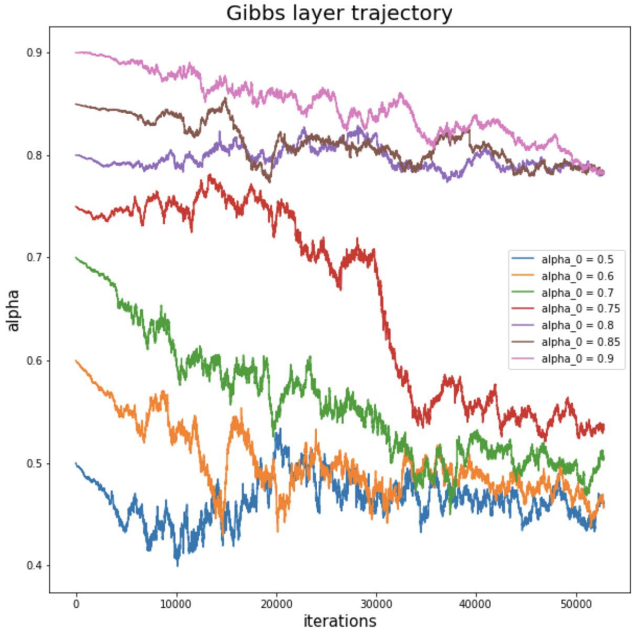

**FIGURE 15.** Trajectories of the Gibbs layers for various starting points.

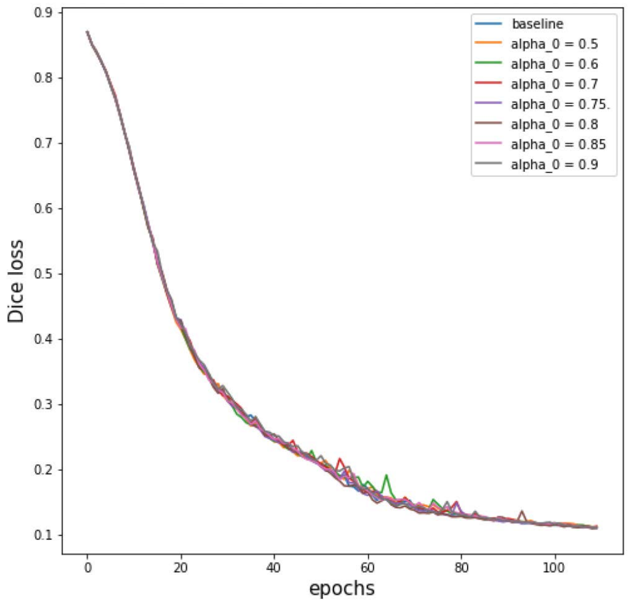

**FIGURE 14.** Learning curves for different implementations of the end-to-end Gibbs ResUnet model. Each loss curve corresponds to a model with a specified starting value for the hyperparameter alpha.

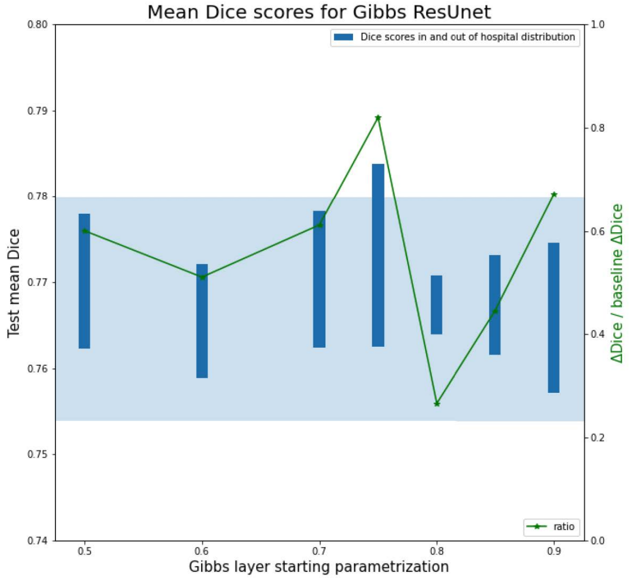

**FIGURE 16.** Inference of Gibbs-stylized models trained using the Gibbs ResUnet. The upper and lower boundaries of the blue band mark the in- and out-of-distribution mean Dice scores for the baseline model, respectively.

- **monai.transforms.KSpaceSpikeNoised** This is a dictionary-based wrapper of KSpaceSpikeNoise. The purpose of this implementation is to allow the user to easily apply different artifact settings to images that are grouped together.
- **monai.transforms.GibbsNoise** This is a deterministic implementation of a filter that applies the Gibbs artifact on the input image.
- **monai.transforms.GibbsNoised** This transform is a dictionary version of the GibbsNoise filter.

The following implementations are probabilistic versions of the filters described above. Instead of taking in descriptors of the artifacts as parameters, they take in ranges from which the descriptors can be sampled. All the transforms listed below inherit probabilistic methods from a superclass, which dictates how the transforms ought to behave.

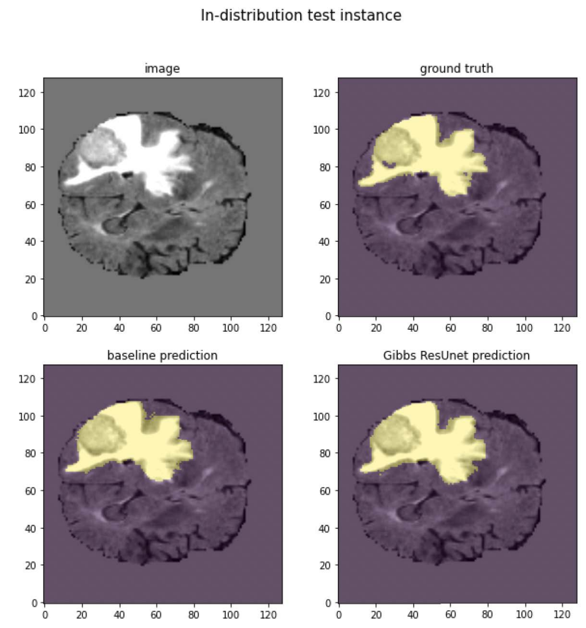

**FIGURE 17.** Example of an in-distribution instance and predicted segmentations. The segmentation maps are overlaid on the image.

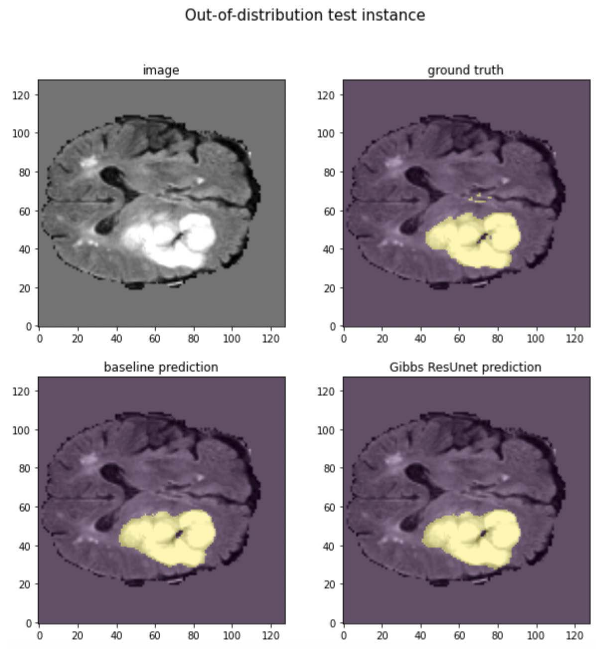

**FIGURE 18.** Example of an out-of-distribution instance and predicted segmentations. The segmentation maps are overlaid on the image.

- **monai.transforms.RandKSpaceSpikeNoise**
- **monai.transforms.RandKSpaceSpikeNoised**
- **monai.transforms.RandGibbsNoise**
- **monai.transforms.RandGibbsNoised**

Finally, we also made public the repository with the source code of the end-to-end Gibbs ResUnet architecture.[3]

## VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we investigated the issue of textural bias exhibited by CNNs, and studied how addressing it can improve robustness to out-of-distribution inference in MRI scans. Specifically, the textural attributes of the images were altered using artifacts commonly seen in clinical MRI images. These artifacts included Gibbs, spike, and wraparound aliasing artifacts, and were all applied in *k*-space.

The initial set of experiments involved training segmentation models on datasets intentionally corrupted with artifacts, and studying how these models performed on new types of artifacts. Our results showed that models trained on certain artifact stylizations were clearly more robust than the baseline model when the intensity the artifact was changed at test time. More importantly, some of these models were indeed able to outperform the baseline model when we changed the type of artifact used in the stylized test set. This leads us to believe that the careful use of artifact stylizations can make the models invariant to low frequency textural features, leading them to learn useful information about the global anatomical structures within the image.

Encouraged by the above observations, we carried out further experiments to explore whether the use of *k*-space artifacts could lead to models performing well under a shift in hospital distribution. Some of the models that were trained on datasets stylized with Gibbs and spikes artifacts exhibited an increased performance on out-of-distribution test sets. These results showed that the stylizations were indeed enabling the models to generalize better by paying less attention to local textural variations, which can be inherent to certain scanner hardware or acquisition protocols. This is an important observation because by using the studied artifacts as pre-processing filters, image segmentation models could be made robust to data acquired from previously-unseen hospitals, making it easier for CNNs to be implemented in routine clinical practice in the long-term.

Lastly, in an effort to automatically obtain the desired improvement on out-of distribution images, we introduced Gibbs ResUnet: a novel, end-to-end framework which incorporates the stylization filter as an initial, learnable layer. Our experiments showed that Gibbs ResUnet can indeed be successfully trained to outperform the baseline model on images coming from a held-out institution.

In terms of efficiency, the 4D and 3D models were trained over approximately 11 and 7 hours respectively when using a single graphics processing unit (GPU). The stylization transforms' runtimes are on par with the other PyTorch preprocessing transforms. This is aided by the fact that the stylization transforms work on batches just like rest of the pipeline and the CNN. When it comes to Gibbs ResUnet, there is the extra step of gradient descent for the Gibbs layer but this means learning only one additional parameter (alpha) out of a total of 4,810,075 parameters. Hence, the runtime of the Gibbs ResUnet is effectively the runtime of the ResUnet.

[3] https://github.com/yanielc/Gibbs_ResUnet

With regards to future work, the findings of this paper point at several interesting questions which could shed more light on the CNN textural bias debate, in the context of MRI. For example, comparing the activations within the best performing stylized networks against those of the baseline model could provide more insight into why the bias is less pronounced in the former, and may help spawn research on inherently unbiased segmentation networks. In the case of the stylizing layers, this work showed that gradient descent can be used to successfully train the end-to-end Gibbs ResUnet architecture. Yet, the stability issue experienced during model training remains unsolved. It would therefore be beneficial to explore whether there is a more stable approach to end-to-end training of the Gibbs layer with the network, e.g. by using alternatives to gradient descent or through pre-training the segmentation component of the model. With regards to the scalability of our findings, the exact degree to which Gibbs ResUnet could aid model robustness in the presence of other MRI artifacts (e.g. chemical shift artifacts, magnetic field inhomogeneities, surface coil artifacts) remains an open question, and investigating this could lead to exciting future research. Lastly, future work should also investigate the initialization of alpha_0 to help Gibbs ResUnet find the ideal balance between robustness and performance degradation.

## ACKNOWLEDGMENT

Ahmed E. Fetit would like to thank the UK Research and Innovation Centre for Doctoral Training in Artificial Intelligence for Healthcare for supporting his research in his role as a Senior Teaching Fellow. The results published here are in whole or part based upon data generated by the TCGA Research Network: http://cancergenome.nih.gov/.

## REFERENCES

[1] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, ''A survey on deep learning in medical image analysis,'' *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[2] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, ''ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness,'' in *Int. Conf. Learn. Represent.*, 2018, pp. 1–22.

[3] N. Kriegeskorte, ''Deep neural networks: A new framework for modeling biological vision and brain information processing,'' *Annu. Rev. Vis. Sci.*, vol. 1, no. 1, pp. 417–446, 2015.

[4] H. Wang and B. Raj, ''On the origin of deep learning,'' 2017, *arXiv:1702.07800*.

[5] J. Kubilius, S. Bracci, and H. P. Op de Beeck, ''Deep neural networks as a computational model for human shape sensitivity,'' *PLOS Comput. Biol.*, vol. 12, no. 4, Apr. 2016, Art. no. e1004896.

[6] W. Brendel and M. Bethge, ''Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet,'' in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.

[7] L. Gatys, A. S. Ecker, and M. Bethge, ''Texture synthesis using convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[8] L. S. Hesse, G. Kuling, M. Veta, and A. L. Martel, ''Intensity augmentation for domain transfer of whole breast segmentation in MRI,'' 2019, *arXiv:1909.02642*.

[9] A. E. Fetit, J. Cupitt, T. Kart, and D. Rueckert, ''Training deep segmentation networks on texture-encoded input: Application to neuroimaging of the developing neonatal brain,'' in *Proc. Med. Imag. Deep Learn.*, 2020, pp. 230–240.

[10] S. Chai, D. Rueckert, and A. E. Fetit, ''Reducing textural bias improves robustness of deep segmentation models,'' in *Proc. Annu. Conf. Med. Image Understand. Anal.* Cham, Switzerland: Springer, 2021, pp. 294–304.

[11] P. Ballester and R. M. Araujo, ''On the performance of GoogLeNet and AlexNet applied to sketches,'' in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1124–1128.

[12] L. A. Gatys, A. S. Ecker, and M. Bethge, ''Texture and art with deep neural networks,'' *Current Opinion Neurobiol.*, vol. 46, pp. 178–186, Oct. 2017.

[13] K. Hermann, T. Chen, and S. Kornblith, ''The origins and prevalence of texture bias in convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 19000–19015.

[14] C. K. Mummadi, R. Subramaniam, R. Hutmacher, J. Vitay, V. Fischer, and J. H. Metzen, ''Does enhanced shape bias improve neural network robustness to common corruptions?'' in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–20.

[15] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng, ''Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation,'' *IEEE Trans. Med. Imag.*, vol. 39, no. 7, pp. 2494–2505, Jul. 2020.

[16] Q. Dou, C. Ouyang, C. Chen, H. Chen, B. Glocker, X. Zhuang, and P.-A. Heng, ''PnP-AdaNet: Plug-and-play adversarial domain adaptation network at unpaired cross-modality cardiac segmentation,'' *IEEE Access*, vol. 7, pp. 99065–99076, 2019.

[17] R. Shaw, C. Sudre, S. Ourselin, and M. J. Cardoso, ''MRI *k*-space motion artefact augmentation: Model robustness and task-specific uncertainty,'' in *Proc. Int. Conf. Med. Imag. Deep Learn.*, 2018, pp. 1–10.

[18] V. Grover, J. M. Tognarelli, M. M. Crossey, I. J. Cox, S. D. Taylor-Robinson, and M. J. McPhail, ''Magnetic resonance imaging: Principles and techniques: Lessons for clinicians,'' *J. Clin. Exp. Hepatol.*, vol. 5, no. 3, pp. 246–255, 2015.

[19] K. T. Block, M. Uecker, and J. Frahm, ''Suppression of MRI truncation artifacts using total variation constrained data extrapolation,'' *Int. J. Biomed. Imag.*, vol. 2008, pp. 1–8, Aug. 2008.

[20] A. L. Simpson *et al.*, ''A large annotated medical image dataset for the development and evaluation of segmentation algorithms,'' 2019, *arXiv:1902.09063*.

[21] L. Scarpace, T. Mikkelsen, S. Cha, S. Rao, S. Tekchandani, D. Gutman, J. H. Saltz, B. J. Erickson, N. Pedano, A. E. Flanders, J. Barnholtz-Sloan, Q. Ostrom, D. Barboriak, and L. J. Pierce, ''Radiology data from the cancer genome atlas glioblastoma multiforme [TCGA-GBM] collection,'' The Cancer Imag. Arch., 2016. [Online]. Available: https://doi.org/10.7937/K9/TCIA.2016.RNYFUYE9, doi: 10.7937/K9/TCIA.2016.RNYFUYE9.

[22] N. Pedano, A. E. Flanders, L. Scarpace, T. Mikkelsen, J. M. Eschbacher, B. Hermes, and Q. Ostrom, ''Radiology data from the cancer genome atlas low grade glioma [TCGA-LGG] collection,'' The Cancer Imag. Arch., 2016. [Online]. Available: http://doi.org/10.7937/K9/TCIA.2016.L4LTD3TK, doi: 10.7937/K9/TCIA.2016.L4LTD3TK.

[23] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, ''The cancer imaging archive (TCIA): Maintaining and operating a public information repository,'' *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, Dec. 2013.

[24] E. Kerfoot, J. Clough, I. Oksuz, J. Lee, A. P. King, and J. A. Schnabel, ''Left-ventricle quantification using residual U-Net,'' in *Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges*. Cham, Switzerland: Springer, 2019, pp. 371–380.

[25] O. Ronneberger, P. Fischer, and T. Brox, ''U-Net: Convolutional networks for biomedical image segmentation,'' in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[26] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, ''Image segmentation using deep learning: A survey,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 17, 2021, doi: 10.1109/TPAMI.2021.3059968.

[27] O. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, ''3D U-Net: Learning dense volumetric segmentation from sparse annotation,'' in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2016, pp. 424–432.

[28] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, Oct. 2020, pp. 1–7.

**YANIEL CABRERA** received the M.Sc. degree in mathematics and the Ph.D. degree in mathematical physics from Texas A&M University, in 2011 and 2017, respectively, and the M.Sc. degree in artificial intelligence from the Imperial College London, in 2021. His research interests include applications of deep learning to computer vision and signal processing.

**AHMED E. FETIT** received the B.Eng. (Hons.) and M.Sc. degrees from the University of Birmingham and the Ph.D. degree from the University of Warwick, where he researched applications of texture analysis and machine learning for the diagnosis and prognosis of pediatrics brain tumors from clinical MRI data. He was a Postdoctoral Researcher at the Biomedical Image Analysis Group, Imperial College London, and the Computer Vision and Image Processing Group, University of Dundee. He is currently a Senior Teaching Fellow with the UKRI Centre for Doctoral Training in Artificial Intelligence for Healthcare, Imperial College London. His research interests include the intersection of biomedical image analysis and machine learning.