Training Improvements for Ultrasound Beamforming with Deep Neural Networks

Adam C. Luchies, Brett C. Byram

***Abstract*—This paper investigates practical considerations of training ultrasound deep neural network (DNN) beamformers. We first present the results of a hyperparameter search based on image quality. The findings show the effect of different hyperparameter settings on final ultrasound image quality and also that final validation loss—normally used to select DNNs—was a poor predictor of image quality. The study also showed that DNN beamforming is robust to electronic noise. Next, we studied image quality in physical phantom and *in vivo* scans and demonstrate that DNN beamformer image quality in these setting was correlated to DNN beamformer image quality in simulated images. This finding suggests that DNN beamformers can be selected based on simulated image quality and that these image quality improvements translate to *in vivo* applications. Finally, we studied the effect of dataset size on DNN beamformer image quality in simulation, physical phantom, and *in vivo* scans. We interpret the results in terms of recent work on the scaling of deep learning. Overall, the results in this paper show that DNN beamforming has significant potential for improving B-mode image quality.**

***Index Terms*—ultrasound beamforming, denoising, neural networks, B-mode image quality**

I. Introduction

THE goal of delay and sum (DAS) beamforming is to apply delays and amplitude weights to received sensor signals such that waves coming from one direction are reinforced during summation relative to waves coming from other directions [1]. Usually these delays and amplitude weights are static and do not depend on the received data. In contrast, adaptive beamformers adjust the beamforming process depending on the received signals [2], [3], [4], [5], [6]. In one type of adaptive imaging, a near field phase screen is used to model wavefront aberration due to sound speed inhomogeneities. After estimation, this phase screen is used to adjust the focusing delays [3]. In another adaptive technique, the covariance of the received signals is used to select the amplitude weights to suppress waves coming from directions other than the look angle [4].

Recently, several approaches were investigated for using deep neural networks (DNNs) for the purposes of ultrasound image reconstruction [7], [8], [9]. One of the primary challenges of this endeavor is that DNN training requires a well labeled training data set, which is difficult to create or obtain for ultrasound channel data. The approach by Gasse et al. was to use a synthetic aperture method (i.e., coherent plane wave compounding) to create an ultrasound channel data set with improved resolution and contrast [10]. DNNs were then trained to produce images with similar image quality improvements, but using many fewer transmit events. The disadvantage of this approach is that image quality is limited by the image quality using synthetic aperture methods.

We developed a method that relied on simulation tools to generate training data [11], [12], [13], [14]. The advantage of using linear ultrasound simulation tools is that they are fast and make it possible to generate as much training data as desired. The disadvantage of using simulation is that the DNNs will be limited by the accuracy of the simulation tool. Our results showed that it was possible to train DNNs using simulations to produce image quality improvements in physical phantoms and *in vivo* scans.

The DNN beamforming method that we developed was inspired by the aperture domain model image reconstruction (ADMIRE) beamforming method that Byram et al. developed [5], [6], [15]. ADMIRE poses beamforming as a nonlinear regression problem and requires a significant amount of computation to perform beamforming. A DNN is a prime candidate for solving nonlinear regression problems and most of the computation is completed during the training phase. The inference phase for DNNs can be fast.

The goal of this work was to use empirical studies to examine practical considerations of training ultrasound DNN beamformers. We show the effect of different hyperparameters on final image quality, study the method for selecting DNN beamformers, and propose evaluating DNN beamformers using simulated images.

II. Methods

We briefly review the DNN beamformer that we developed in the past and also discuss several innovations [11].

A. Frequency domain processing

The first step in the signal path was to convert the channel data to the frequency domain using a short-time Fourier transform (STFT). The STFT gate length was 16 samples (approximately one pulse length) and a 16 point DFT was used. The gate overlap was 90% and a rectangular window was used. A diagram showing the frequency domain processing by a set of DNNs is in Figure 1.

B. Neural Networks

A DNN beamformer is a set of DNNs to process data for different DFT bins. In this work, a DNN beamformer consisted of three DNNs, each trained to process a different DFT bin.

The authors are with the Department of Biomedical Engineering, Vanderbilt University, Nashville, TN, 37212 USA.

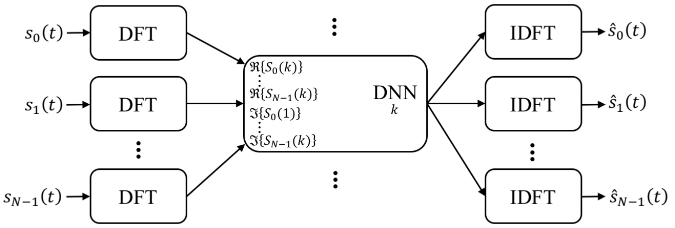

Fig. 1. Diagram showing frequency domain processing by a set of DNNs indexed by frequency, k. N is the number of elements in the receive subaperture. $s_n(t)$ is the gated data for the nth channel. The input signals are a single gated depth of channel data. A discrete Fourier transform (DFT) transformed each channel signal into the frequency domain. $\Re$ indicates the real component and $\Im$ indicates the imaginary component. Processed frequency domain data is transformed back into the time domain using an inverse discrete Fourier transform (IDFT) [16].

TABLE I
HYPERPARAMETER SEARCH SPACE

Parameter	Search Values
Training Dataset #	1,2,3
Batch Size	50,100,500,1000
Number of Hidden Layers	1-5
Layer Width	65,130,260,520
Input Gaussian Noise	True or False
Input Dropout Probability	0, 0.1, 0.2
Hidden Node Dropout Probability	0, 0.1, 0.2, 0.3, 0.4, 0.5
L2 Weight Decay	$0, 10^{-5}, 10^{-4}$

The DFT bins were for the transmit center frequency and the adjacent DFT bin on either side of this center frequency. For the pulse-echo characteristics specified, all other DFT bins were 20 dB below that of the center frequency. The remaining DFT bins were zeroed out when transforming back to the time domain, which is an approach used elsewhere [6], [15].

We trained fully connected feed-forward multilayer networks to process aperture data and performed a hyperparameter search with parameters indicated in Table I. The hyperparameters were specified on a grid and random configurations of hyperparameters were selected for creating DNN beamformers. For a single DNN beamformer, all DNNs were trained with the same hyperparameters and model architecture. A total of 100 DNN beamformers were trained.

Gradient descent is by far the most popular algorithm for training neural networks [17]. In addition, the learning rate is widely viewed as the most important hyperparameter to tune when training neural networks [18]. Adaptive learning rate methods, such as Adam (adaptive moment estimation), adjust the learning rate based on parameters themselves. The learning rate is increased for parameters with infrequent or small valued gradients and decreased for parameters with frequent or large valued gradients. We used Adam with the values suggested by Kingma et al. ($\alpha = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) [19].

The DNNs used the rectified linear unit (ReLU) for the activation function [20]. The weights of the network were initialized using the probability distribution given by [21], [22]

$$W \sim N\left(0, \sqrt{\frac{2}{n}}\right) \tag{1}$$

where $N(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2, and n is the size of the previous layer. This probability distribution was developed based on expected activation rates when using the ReLU activation [22]. Initializing weights in this manner avoids reducing or magnifying input signal magnitudes as signals propagate through the network and improves network convergence. The biases for each neuron were initialized to a value of 0.01. Training was terminated if the validation loss did not improve after 20 epochs. During training, networks were only saved if the validation loss decreased. Thus, the final set of network weights were the ones that produced the best validation loss for a training run.

We examined using different levels of dropout probability for the hidden layers as indicated in Table I [23]. The dropout probability used for inputs is usually smaller than that used for the hidden nodes. The input dropout probability values are in Table I. We studied using different amounts of $L2$ weight decay as indicated in Table I. Input Gaussian noise was also enabled or disabled to study whether input noise improved DNN beamformers. When enabled, white Gaussian noise was added to batches of training and validation data with variable SNR in the range 0-40 dB. These methods have the effect of regularizing a DNN, can prevent a network from overfitting (i.e., memorizing the training data), and aid in network generalization.

During training, DNN inputs were normalized such that individual input vectors had maximum norm equal to one. During the inference phase, the input vector was normalized by its maximum norm and the DNN output was renormalized by the input maximum norm.

Pytorch was used to create and train all of the DNNs in this work [24]. Training was performed on a GPU computing cluster maintained by the Advanced Computing Center for Research and Education at Vanderbilt University.

C. Training Data

Three datasets were created and used for training DNN beamformers. The three datasets were treated as a hyperparameter as in Table I. Dataset #1, dataset #2, and dataset #3 are described in Fig. 2 (a), (b), and (c) respectively. Note that dataset #1 had the same configuration as our previous work on training DNN beamformers [11]. In all methods, scatterers were randomly placed in an annular sector as depicted in Fig. 2. The annular sector had a width of 50 pulse lengths. The responses from a particular scatterer was only kept if it appeared in the STFT gate that was centered at the transmit focus. The goal was to train a network for this STFT gate.

The scatterers were divided into two groups, including on-axis and off-axis scatterers. The division was set as the region between the nulls of the main lobe for a simulated beam at the transmit center frequency. For on-axis scatterers, the target signal was the same as the input signal. For off-axis scatterers, the target signal was a vector of zeros. When an input signal was the combination of on-axis and off-axis scatterers as in datasets #2 and #3, the target consisted of the summation of on-axis signals. If the input consisted of the responses from only off-axis scatterers, the target was a vector of zeros.

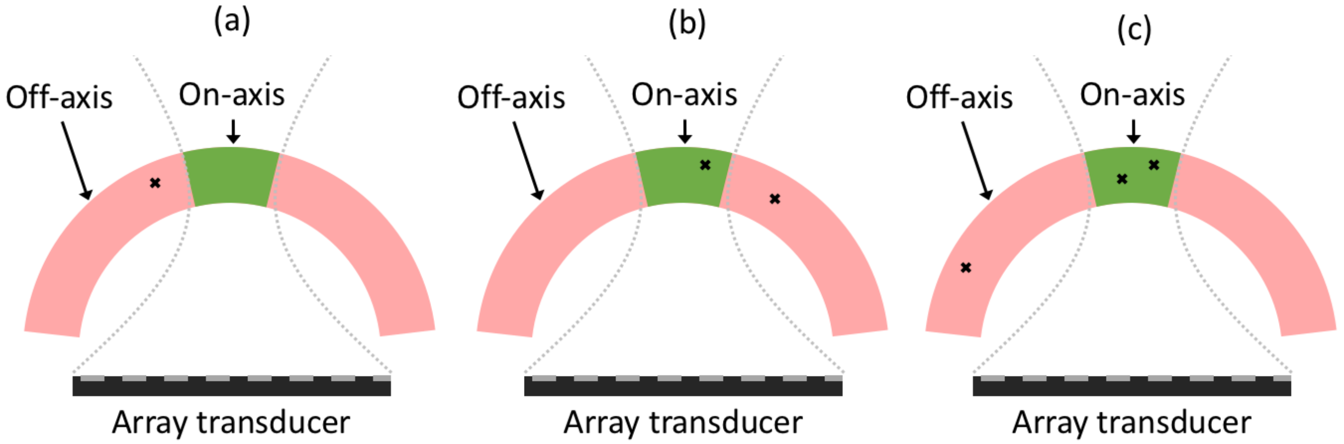

Fig. 2. (a) Dataset #1 consisted of the responses from single scatterers. (b) One half of dataset #2 consisted of responses from a single scatterer (similar to dataset #1) and the second half consisted of the combined responses from two scatterers at different locations. (c) One third of dataset #3 consisted of responses from a single scatterer (similar to dataset #1), the second third consisted of the combined responses from two scatterers at different locations (similar to dataset #2), and the final third consisted of the combined responses from scatterers at three different locations.

TABLE II
LINEAR ARRAY SCAN PARAMETER VALUES

Parameter	Value
Active Elements	65
Transmit Frequency	5.208 MHz
Pitch	298 μm
Kerf	48 μm
Simulation Sampling Frequency	520.8 MHz
Experimental Sampling Frequency	20.832 MHz
Speed of Sound	1540 m/s
Transmit Focus	70 mm

All training data was generated using Field II. The simulation parameters are in Table II. The simulated transducer was modeled after an ATL L7-4 (38 mm) linear array transducer. For each dataset described in Fig. 2, a training and a validation dataset were created. The training dataset was only used to adjust weights. The validation dataset was used to stop training and studied as a method for DNN beamformer selection. The training datasets had 100,000 examples and the validation datasets had 10,000 examples.

D. Image Quality Metrics

We quantified image quality using contrast ratio (CR)

$$\mathrm{CR} = -20\log_{10}\left(\frac{\mu_{\text{lesion}}}{\mu_{\text{background}}}\right), \tag{2}$$

contrast-to-noise ratio (CNR) [25], [26], [27]

$$\mathrm{CNR} = 20\log_{10}\left(\frac{|\mu_{\text{background}} - \mu_{\text{lesion}}|}{\sqrt{\sigma^2_{\text{background}} + \sigma^2_{\text{lesion}}}}\right), \tag{3}$$

and speckle signal-to-noise ratio (SNRs)

$$\mathrm{SNRs} = \frac{\mu_{\text{background}}}{\sigma_{\text{background}}}, \tag{4}$$

where μ is the mean and σ is the standard deviation of the uncompressed envelope. When estimating electronic SNR between successive frames, we used [28]

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\rho}{1-\rho}\right), \tag{5}$$

where ρ is the correlation coefficient between two successive frames.

E. Simulation: Channel SNR Analysis

We studied the robustness of DNN beamformers to noise using FIELD II simulations [13] [14]. In each simulation, an anechoic cyst at a depth of 7 cm and having a diameter of 5 mm was imaged using a simulated L7-4 38 mm linear array. No scatterers were located inside the cyst and 25 scatterers per resolution cell were placed in the background region. The noiseless channel data was then corrupted using white Gaussian noise to vary channel SNR from 10 dB to -10 dB in steps of 5 dB. A total of five simulated anechoic cysts were studied.

F. Physical Phantom and In Vivo Scans

A linear array transducer (ATL L7-4 38 mm) was operated using a Verasonics Vantage 128 system (Verasonics, Kirkland, WA) to conduct physical phantom and *in vivo* scans. The physical phantom was a multipurpose phantom (Model 040GSE, CIRS, Norfolk, VA) and a cylindrical anechoic cyst at a 7 cm depth with an approximately 3 mm diameter was scanned. Five scans were made at different positions along the cylindrical cyst.

The same ultrasound system and transducer were used to scan the liver of a 36 year old healthy male. Scanning was conducted to look at liver vasculature. The study was approved by the local Institutional Review Board.

G. The Effect of Training Dataset Size

When training DNNs, one of the most reliable approaches for performance improvement is to increase training dataset size [29]. Studying model performance as a function of this quantity provides a way to predict potential performance improvements using this strategy.

After selecting a DNN beamformer based on simulated image CNR, we retrained this DNN beamformer using the same hyperparameter settings and model architecture multiple times to study the effect of training dataset size. The studied training dataset size include 10^2, 10^3, 10^4, and 10^5. For each training dataset size, the DNN beamformer was retrained five times using different starting weights. In each case, the size of the validation dataset was kept constant with a value of 10^4.

III. Results

A. Channel SNR Analysis

1) Hyperparameters and Image Quality: Fig. 3 shows image quality for simulated anechoic cysts as a function of channel SNR for different hyperparameters. Each row demonstrates the effect of a different hyperparameter on image quality. For example, Fig. 3 (a-c) shows image quality for three DNN beamformers. The first DNN beamformer was selected from the group of beamformers that were trained using dataset #1 and it was selected as the beamformer that had the best performance in terms of CNR. The results for this beamformer are shown as the blue (circle) line. The second DNN beamformer was selected from the group of beamformers that were trained using dataset #2 and it was selected as the beamformer that had the best performance in terms of CNR. The results for this beamformer are shown as the orange (triangle) line. The third DNN beamformer was selected from the group of beamformers that were trained using dataset #3 and it was selected as the beamformer that had the best performance in terms of CNR. The results for this beamformer are shown as the green (square) line. Displaying results in this manner provides a way to visualize the effect of different hyperparameters on final image quality, while marginalizing out the remaining hyperparameters.

Fig. 3 (b) and (c) show that training with datasets #2 and #3 produced the best CNR for lower channel SNR and the best speckle SNR for all channel SNR values. However, dataset #1 produced the best CR for higher channel SNR. These results demonstrate that training with the responses from multiple point targets improved speckle SNR compared to training responses from a single point target. However, for high channel SNR, dataset #1 performed just as well in terms of CNR.

Fig. 3 (e) and (f) show the effect of batch size on image quality. Better CNR was produced when channel SNR was lower by using a larger batch size. In addition, the best speckle SNR was produced in general by using a batch size equal to 1000. It should be noted that when disabling network regularization methods such as dropout or weight decay, using smaller batch sizes produced the best results in terms of image quality.

Figs. 3 (g-i) show how image quality improved as the number of hidden layers increased. Similar to most deep learning applications, increasing depth improved performance for DNN beamforming. It should be noted that the network with five hidden layers produced lower image quality than did the network with four hidden layers. Increasing the training data set size would likely reverse this trend. Fig. 3 (k) and (l) show similar trends in that using networks with wider layers improved image quality.

Figs. 3 (m) show that there was little difference for CNR and speckle SNR when enabling or disabling input Gaussian noise. In contrast, Figs. 3 (p-r) show that image quality improved when using input dropout.

Figs. 3 (s-u) show that using some dropout for the hidden layers improved image quality. This finding is in contrast to our previous work which suggested that dropout was not advantageous for training networks for DNN beamforming [11]. However, in that work we showed DNN beamformer training results in terms of validation loss and not final image quality. Figs. 3 (v-x) show that enabling weight decay did not provide any benefit for image quality compared to disabling weight decay.

Overall, Fig. 3 shows that the trained DNN beamformers were robust to noise. The improvement in CNR of DNN beamformers compared to DAS actually increased for higher levels of noise.

2) Validation Loss and Image Quality: Typically, DNNs are selected based on final validation loss. However, Fig. 4 shows that network validation loss was not a good predictor of final image quality. The R^2 for Figs. 4 (a), (b), and (c) were -0.05, -0.62, and -0.12, respectively. The best DNN beamformer in terms of CNR was trained with dataset #2 and the final validation loss was an order of magnitude higher than the DNN beamformer with the lowest validation loss. The DNN beamformer with lowest validation loss produced CNR that was 15-30% lower than the DNN beamformer with best CNR. Fig. 4 (a) also shows how the CNR variance amongst DNN beamformers that used dataset #1 was larger than the CNR variance for DNN beamformers that used datasets #2 and #3.

3) Image Quality Tradeoffs: Fig. 5 shows scatter plot relationships between the different image quality metrics for the trained DNN beamformers. Fig. 5 (a) shows a parabolic relationship between CR and CNR. Fig. 5 (b) shows a linear relationship between speckle SNR and CNR when speckle SNR was less than 1.7. However, the variance of CNR as a function of speckle SNR increased for higher values of speckle SNR. Fig. 5 (c) shows a negative linear relationship between speckle SNR and CR. It is noteworthy that many of the DNN beamformers produced better CR and CNR compared to DAS. In addition, many of the DNN beamformers produced speckle SNR values that were within 10% of the speckle SNR for DAS.

4) Channel SNR Improvements: Fig. 6 shows that the trained DNN beamformers increased channel SNR by at least 10 dB and up to 20 dB compared to the input channel SNR. These results provide evidence of the denoising ability of DNN beamformers.

5) DNN Beamformer Selection and Performance: We selected the best performing DNN beamformer by averaging the measured CNR values for the studied channel SNR values and picking the DNN beamformer with the best average CNR. The hyperparameter values for the DNNs in this beamformer were as follows: the DNNs were trained using dataset #2, 500 batch size, 4 hidden layers, no Gaussian noise was added to the

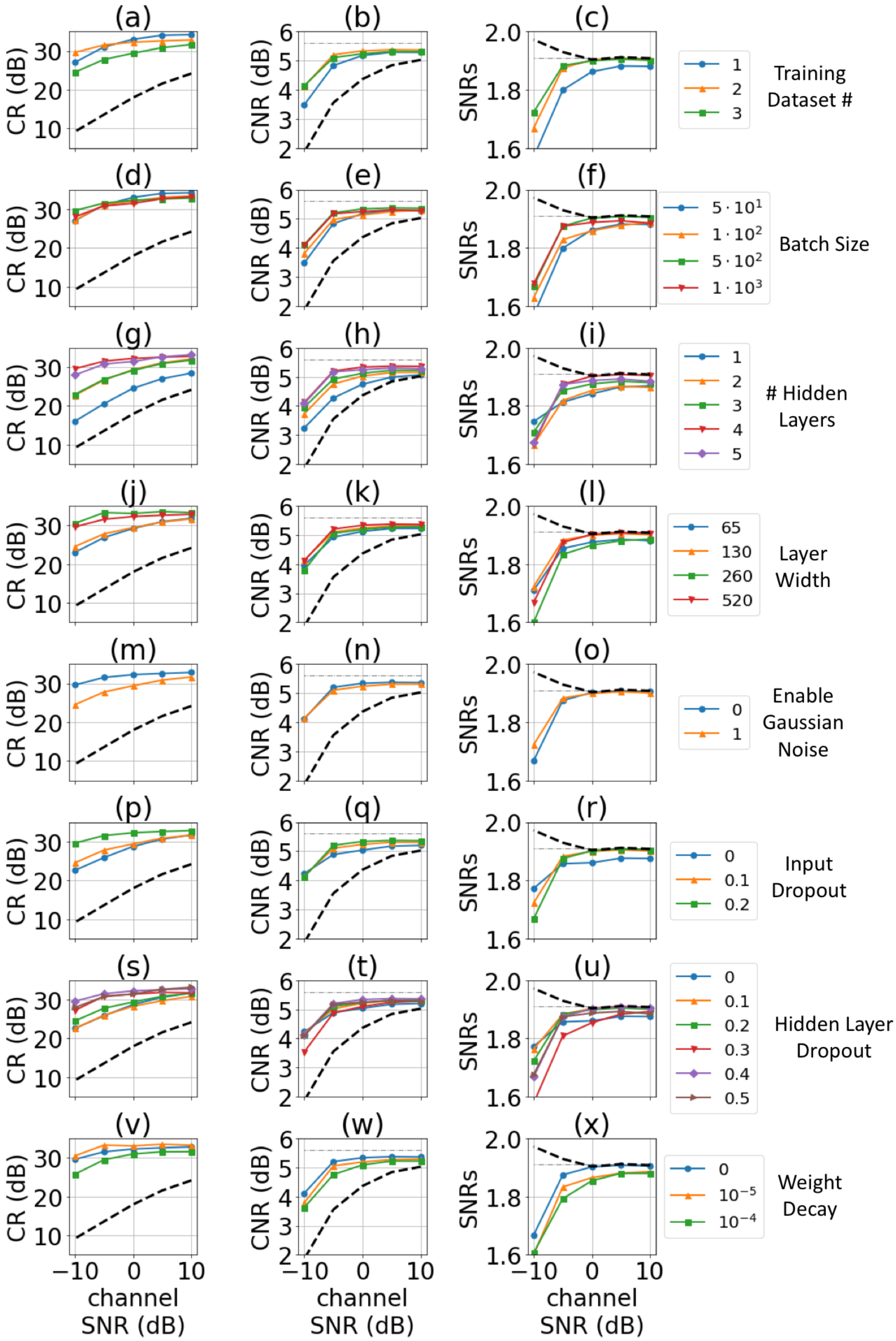

Fig. 3. CR, CNR, and speckle SNR for DNN beamformers as a function of channel SNR. (a-c) Out of all DNN beamformers that used training dataset #1, we selected the best DNN beamformer in terms of CNR, and display CR, CNR, and SNRs for this beamformer as the blue (circle) line. Out of all DNN beamformers that used training dataset #2, we selected the best DNN beamformer in terms of CNR, and display CR, CNR, and SNRs for this beamformer as the orange (triangle) line. Out of all DNN beamformers that used training dataset #3, we selected the best DNN beamformer in terms of CNR, and display CR, CNR, and SNRs for this beamformer as the green (square) line. The remaining rows show the best performing DNN beamformer in terms of CNR for the indicated hyperparameter value. For comparison, the dashed black line shows the performance of DAS. The light gray dashed line shows the theoretical limit for CNR.

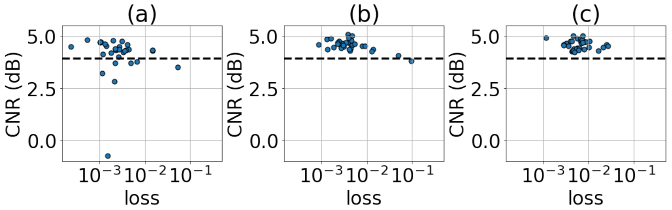

Fig. 4. CNR as a function of validation loss for DNN beamformers trained with (a) dataset #1, (b) dataset #2, and (c) dataset #3. The validation loss is the average validation loss for all DNNs in a DNN beamformer. The CNR values are average values across the channel SNR range studied in the anechoic cyst simulation. The dashed line indicates CNR for DAS.

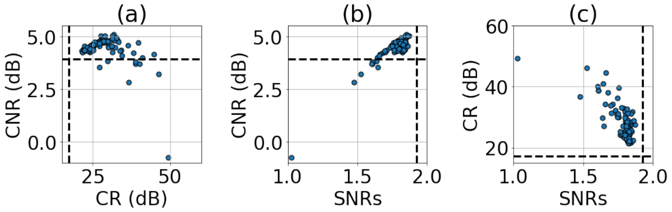

Fig. 5. Scatter plots for (a) CR and CNR, (b) speckle SNR and CNR, and (c) speckle SNR and CR. The speckle statistic values are average values across the channel SNR range studied in the anechoic cyst simulation. The dashed lines indicate performance achieved by DAS.

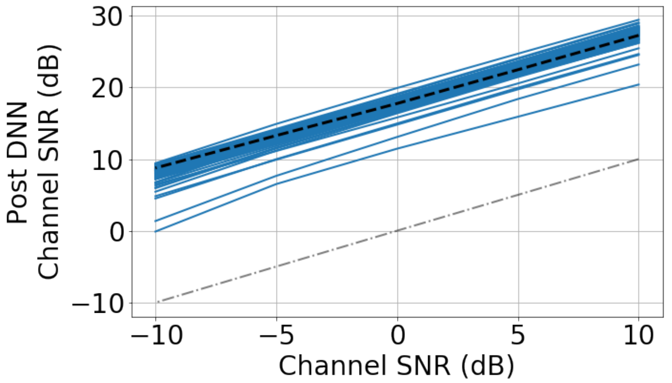

Fig. 6. Each blue line represents the channel SNR after processing by a specific a DNN beamformer. The dashed black line is the channel SNR for the best DNN beamformer selected based on CNR. The dashed-dotted gray line is the input channel SNR.

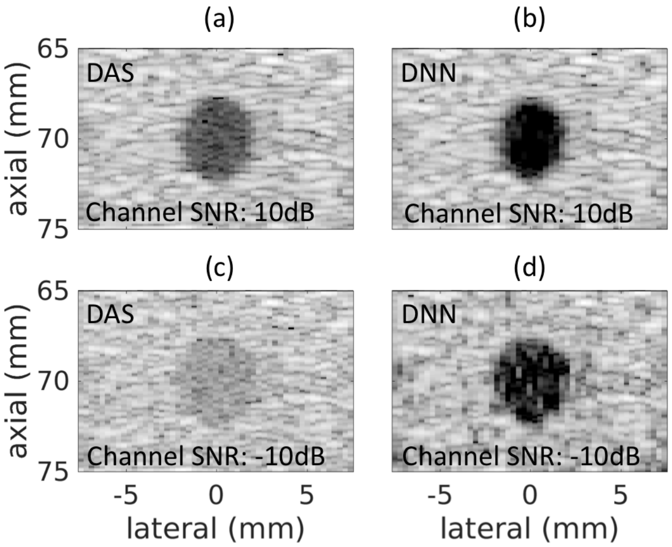

Fig. 7. DAS and DNN images for low and high channel SNR. Images shown with 60 dB dynamic range.

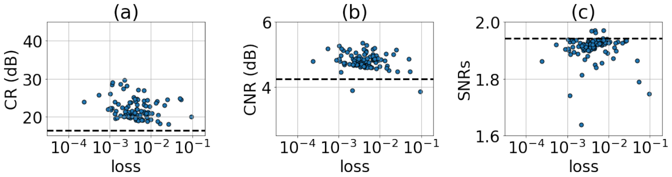

Fig. 8. (a) CR, (b) CNR, and (c) SNRs for an anechoic cyst inside a physical phantom as a function of validation loss. The validation loss is the average validation loss for all DNNs in a DNN beamformer. The dashed line indicates CR, CNR, and SNRs for DAS.

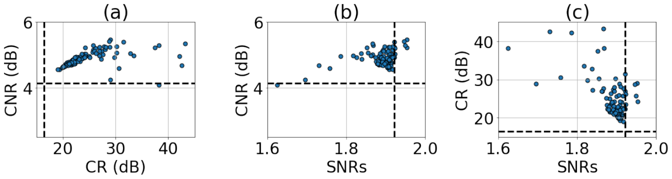

Fig. 9. Scatter plots for (a) CR and CNR, (b) speckle SNR and CNR, and (c) speckle SNR and CR for an anechoic cyst in a physical phantom. The dashed lines indicate performance achieved by DAS.

inputs, 0.2 input dropout probability, 0.4 hidden layer dropout probability, and no weight decay was used. Simulated images of anechoic cysts using this DNN beamformer are in Fig. 7, which demonstrate the robustness of DNN beamforming to electronic noise along with the improvements to image quality offered by DNN beamforming.

B. Physical Phantom

1) Validation Loss and Image Quality: Fig. 8 shows that validation loss was also not a good predictor for image quality in the physical phantom scans. The R^2 for Figs. 8 (a), (b), and (c) were -0.17, -0.02, and 0.05, respectively.

2) Image Quality Tradeoffs: Fig. 9 shows DNN beamformer scatter plot relationships between the different image quality metrics on physical phantom scans. Similar trends were observed compared to Fig. 5. For example, Fig. 9 (a) shows an approximately parabolic relationship between CR and CNR. Fig. 5 (b) shows a linear relationship between speckle SNR and CNR when speckle SNR was less than 1.7, after which point the variance of CNR increased. Fig. 5 (c) shows a negative linear relationship between speckle SNR and CR. Similar to the simulation results, Fig. 9 show that most DNN beamformers produced better CR and CNR compared to DAS. In addition, many of the DNN beamformers produced speckle SNR values that were within 10% of the speckle SNR for DAS, with some of them also producing larger speckle SNR values than DAS.

3) DNN Beamformer Performance: Fig. 10 provides a comparison of DAS and DNN beamformer images when using the DNN beamformer with the best CNR on simulated images. Contrast improvements are noticeable in the DNN

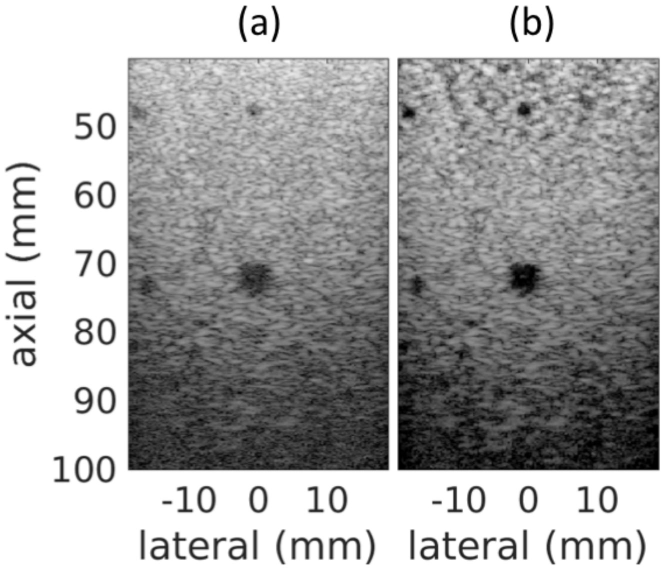

Fig. 10. (a) DAS and (b) DNN images for anechoic cyst in a physical phantom. Images shown with 60 dB dynamic range.

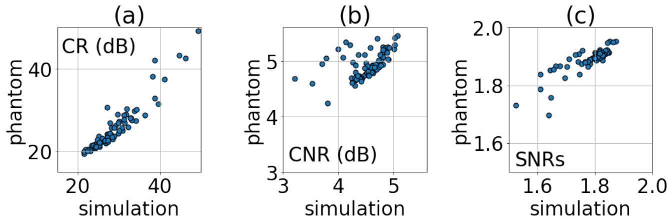

Fig. 11. Scatter plots for image quality measured on simulated data and on physical phantom data. Each circle represents an individual DNN beamformer.

beamformer image. After a depth of about 55 mm, the speckle pattern of the DNN beamformer image is identitcal to that of DAS. At shallower depths, the speckle variance of the DNN beamformer image increased. However, contrast for the small anechoic cysts at 50 mm depth also improved in the DNN beamformer image compared to the DAS image.

It should also be noted that the DNN beamformer was trained assuming a beam centered in the active subaperture. In the lateral regions of the scan in Fig. 10 (i.e., less than -1 cm and greater than 1 cm), the beams were not centered in the active subaperture. For the DNN beamformer, speckle variance appeared to increase slightly in these regions compared to DAS; however, training DNN beamformers to account for non-centered beams should reduce this artifact in these regions.

4) Image Quality in Simulation and Phantom Scans: Fig. 11 shows scatter plots for image quality metrics measured in simulation compared to image quality metrics measured in a physical phantom. The R^2 coefficient for CR, CNR, and speckle SNR was 0.95, 0.85, and 0.94, respectively. In general these results show that for the studied DNN beamformers, the image quality measured in simulation correlated with the image quality measured in the physical phantom. This finding suggests that selecting DNN beamformers based on simulated images is a better strategy than selecting based on final validation loss.

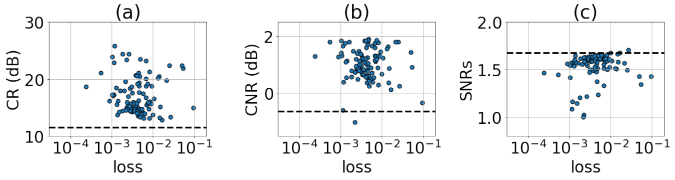

Fig. 12. (a) CR, (b) CNR, and (c) SNRs for *in vivo* scans as a function of DNN beamformer validation loss. The validation loss is the average validation loss for all DNNs in a DNN beamformer. The dashed lines indicate CR, CNR, and SNRs for DAS.

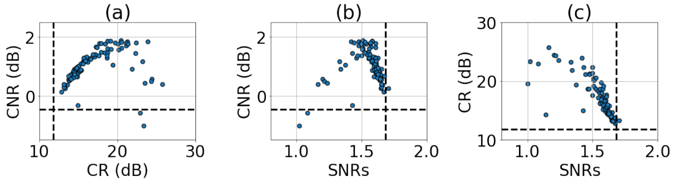

Fig. 13. Scatter plots for (a) CR and CNR, (b) speckle SNR and CNR, and (c) speckle SNR and CR for *in vivo* scans. The dashed black lines indicate image quality for DAS.

C. In vivo

1) Validation Loss and Image Quality: Fig. 12 shows that validation loss was not a good predictor for final image quality in *in vivo* scans, which is consistent with simulation and physical phantom scans. The R^2 for Figs. 12 (a), (b), and (c) were -0.07, 0.03, and 0.18, respectively.

2) Image Quality Tradeoffs: Fig. 13 shows scatter plot relationships between the different image quality metrics for the trained DNN beamformers for *in vivo* scans. The trends in Fig. 13 were similar to those observed in simulated and physical phantom scans. However, Fig. 13 shows that for speckle SNR values greater than 1.5, CNR decreased. In addition, Fig. 13 show that most DNN beamformers produced better CR and CNR compared to DAS for *in vivo* scans.

3) DNN Beamformer Performance: Fig. 14 provide comparison images for DAS and DNN beamforming. The DNN beamformer used was selected as the DNN beamformer with best CNR in simulated images. These results show the increase in contrast when using DNN beamforming. Fig. 14 (a) also shows the speckle preserving ability of DNN beamformers. Similar depths of field are also apparent for the DNN beamformer in physical phantom and *in vivo* images when comparing Fig. 10 and Fig. 14.

4) Image Quality in Simulation and In vivo Scans: Fig. 15 shows scatter plots for image quality metrics measured in simulation compared to image quality metrics measured in *in vivo* scans. The R^2 coefficient for CR, CNR, and speckle SNR was 0.84, 0.84, and 0.78, respectively. In general these results show that for the studied DNN beamformers, the image quality measured in simulation correlated with the image quality measured in the *in vivo* scans. This finding provides further evidence that selecting DNN beamformers based on simulated images is a better strategy than selecting based on final validation loss.

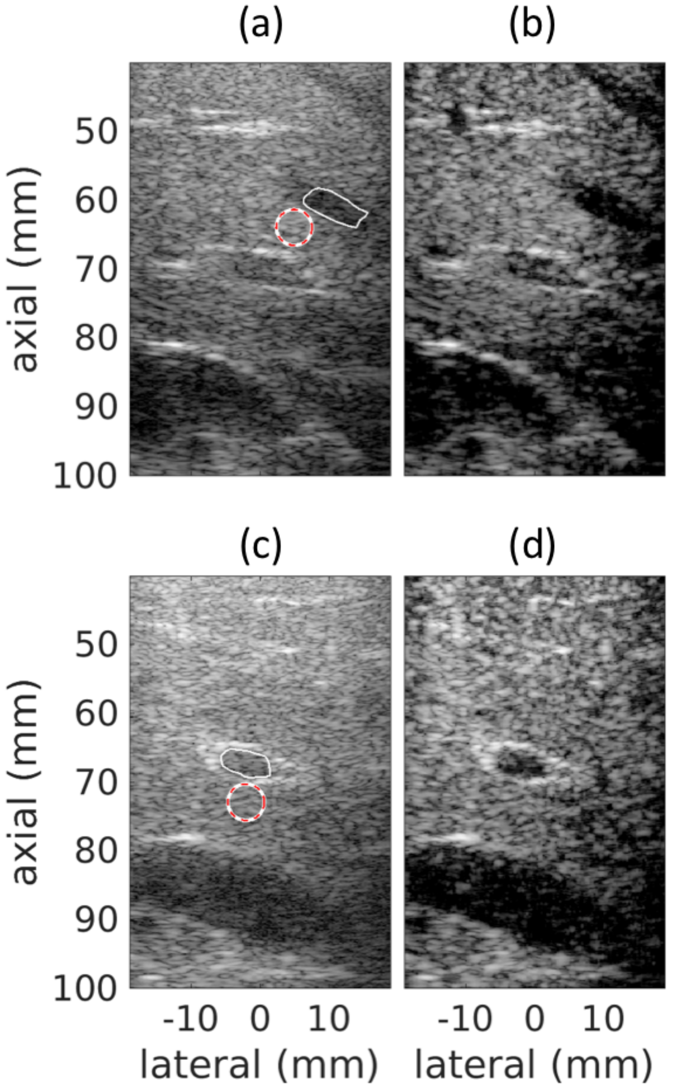

Fig. 14. *In vivo* scans of human liver using (a, c) DAS and (b, d) DNNs. Images shown with 60 dB dynamic range.

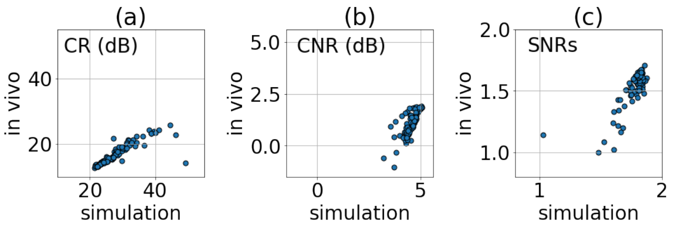

Fig. 15. Scatter plots for image quality measured on simulated data and *in vivo*. Each circle represents an individual DNN beamformer.

D. Training Dataset Size

Fig. 16 shows the effect of training dataset size on image quality for simulation, physical phantom, and *in vivo* scans. In particular, there were large improvements in CR and CNR between dataset sizes of 10^3 and 10^4 for simulation, physical phantom, and *in vivo* scans. In addition, the variance amongst DNN beamformers trained at the same dataset size decreased significantly for training dataset sizes equal to or greater than 10^4 in physical phantom and *in vivo* scans. These findings show that DNN beamformers trained with the same settings but with different weight initialization produced consistent image quality as long as the training dataset size was large enough.

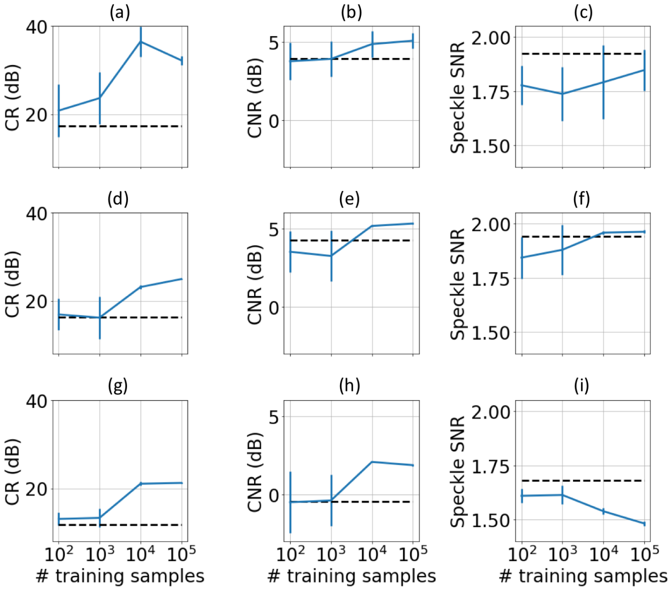

Fig. 16. Image quality as a function of training sample size for (a-c) simulation SNR study, (d-f) physical phantom, and (g-i) *in vivo* scans. The dashed lines indicate performance achieved by DAS.

IV. Discussion

If on-axis scatterering is considered the signal of interest and off-axis scatterering a perturbation that needs to be removed, the DNNs in this work can be viewed as being similar to denoising autoencoders [30]. Autoencoders have several uses including dimensionality reduction, representation learning, and deep neural network pre-traning [18]. Interpreting DNN beamformers as denoising autoencoders also allows us to hypothesize that the DNN beamformers operate by learning an encoding stage, perturbation suppression stage, and decoding stage. Compared to the input domain, the hidden representation provides a better space to suppress the unwanted perturbation (i.e., off-axis scattering and noise) while preserving the signal of interest (i.e., on-axis scattering). It should be noted that methods like ADMIRE can also be interpreted as an encoder, perturbation suppression, and decoder.

Figs. 5, 8, and 12 demonstrate the wide range of image quality achieved by the different DNN beamformers for simulated, physical phantom, and *in vivo*. Overall, almost all DNN beamformers improved image contrast. However, some of them also caused severe speckle degradation. The DNN beamformers that performed well in terms of image quality probably learned the best hidden representation for suppressing off-axis scattering.

The DNN beamformer operates on channel data in the frequency domain. The advantage of operating on frequency domain data is that the beamformer is less dependent on pulse characteristics such as center frequency, bandwidth, and phase. Frequency dependent attenuation changes these pulse characteristics as an acoustic pulse propagates in soft tissue. It should be possible to train DNN beamformers to operate on time-domain data, but doing so would require accounting for these pulse characteristic variations.

Fig. 3 (a-c) show that including the combined responses from multiple scatterers increased speckle SNR. However, using the responses from single scatterers produced the best CR. For higher channel SNR, using the combined responses from multiple scatterers made little difference for CNR. However, for lower channel SNR, using combined responses from multiple scatterers produced better CNR. In addition, the best training dataset was dataset #2. We would expect that as model size and training data size increase, datasets that include the combined responses from larger numbers of scatterers will continue to show a trend of improved image quality.

Figs. 4, 8, and 12 show that validation loss was not a good predictor for final ultrasound image quality. This finding motivates a search for new loss functions for training DNN beamformers.

Many advanced beamforming methods are rigid in the sense that they have a few tuning parameters that can be adjusted to tradeoff different types of image quality improvements. In contrast, DNN beamformers are flexible and are limited by model capacity and training dataset design. The DNN beamformers reported here were designed to suppress off-axis scattering. In theory, it should be possible to design DNN beamformers to suppress other sources of image degradation such as phase aberration or reverberation clutter, and doing so would amount to tweaking the training dataset. As another example, one could envision designing training datasets to improve resolution. Whereas other beamformers might be developed for and work well for specific applications such as calcification detection or correcting phase aberration, DNN beamformers could be targeted to almost any application by modifying the training dataset.

Fig. 3 shows the result that enabling input white Gaussian noise did not offer improvement compared to disabling input white Gaussian noise. We suspect these results can be explained because the DNN beamformers were trained to remove a perturbation (i.e., off-axis scattering) from a signal of interest (i.e., on-axis scattering) and this removal process encouraged the DNNs to be able to handle other sorts of perturbations such as white Gaussian noise. However, it is curious that using input dropout did offer image quality improvements. Training with input dropout in this manner suggests that DNN beamformers could be trained to improve image quality when there are missing or blocked elements [31], [32].

Adaptive gradient descent algorithms offer faster convergence to minima and remove the need to design a learning rate annealing schedule. However, Wilson et al. performed empirical studies that showed that adaptive methods found worse minima in terms of generalization compared to vanilla SGD [33]. Keskar et al. discuss that model parameters that settle in a flatter minima also generalize better and that one method to find these improved minima is to use vanila SGD with a small batch size and large learning rate [34]. It would be interesting to study image quality of DNN beamformers as a function of the SGD variant used for training, but we save this for a future work.

Hestness et al. note that the deep learning community has relied on three recipes for advancing deep learing performance, including searching for improved model architectures, creating larger training data sets, and scaling computation [29]. They note that model architecture advances tend to be serendipitous, whereas the other recipes more reliably improve performance. In addition, they also postulate a power law learning curve as a function of training dataset size and note that in many applications, model architectures merely shift the intercept of this power law and do not change the exponent. In other words, the rate of performance improvement as a function of model size and training dataset size appears to be the same for differing model architectures.

Hestness et al. argue that DNN learning curves can be divided into three regions [29]. First, there is a flat region called the small data region where DNN performance is poor and constant as a function of training dataset size. Second, is the power law region where performance increases linearly as a function of training dataset size. Finally, there is the irreducible error region, where the performance curve is flat again with increasing training dataset size.

Focusing on Fig. 16 (d), a small data region is apparent for training dataset sizes between 10^2 and 10^3 and CR is comparable to DAS. Next, between 10^3 and 10^5, a power law region is apparent as CR improves dramatically compared to DAS. The findings suggest that further improvements in CR might be achieved using larger training datasets. Improving CR is necessary to expand the contrast ratio dynamic range of DNN beamformers [35]. Doing so could enable DNN beamformers to avoid or reduce the dark artifact region that affects many adaptive beamforming methods [36].

V. Conclusion

In this work, we report on the noise robustness of DNN ultrasound beamforming. We used simulations to demonstrate that DNN beamformers improved image quality in low and high noise situations and that DNN beamformers increased channel SNR by 10-20 dB. In addition, we studied the effect of different hyperparameters on ultrasound image quality. We also demonstrated that the final validation loss of DNN beamformers was not a good predictor for final image quality, which motivates trying to identify a new loss function in the future. Based on these findings, we studied image quality for all of the trained DNN beamformers in physical phantom and *in vivo* scans and showed that the final image quality in physical phantom scans and *in vivo* scans correlated with simulation image quality. These results suggest that selecting a DNN beamformer based on simulated image quality is a better strategy than selecting based on final validation loss. Overall, the results in this report demonstrate the potential of using DNN beamformers to improve B-mode ultrasound image quality.

Acknowledgment

The authors would like to acknowledge NIH Grants R01EB020040 and S10OD016216-01. The authors also acknowledge the staff of the Advanced Computing Center for Research and Education at Vanderbilt University.

References

[1] D. H. Johnson and D. E. Dudgeon. *Array signal processing: concepts and techniques*. Prentice Hall, 1993.

[2] P. C. Li and M. L. Li, ”Adaptive imaging using the generalized coherence factor,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 50, no. 2, pp. 128–141, 2003.

[3] J. .J. Dahl, S. A. McAleavey, G. F. Pinton, M. S. Soo, and G. E. Trahey, ”Adaptive imaging on a diagnostic ultrasound scanner at quasi real-time rates,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 53, no. 10, pp. 1832–1843, 2006.

[4] I. K. Holfort, F. Gran, and J. A. Jensen, ”Broadband minimum variance beamforming for ultrasound imaging,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 56, no. 2, pp. 314–325, 2009.

[5] B. Byram and M. Jakovljevic, ”Ultrasonic multipath and beamforming clutter reduction: A chirp model approach,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 61, no. 3, pp. 428–440, 2014.

[6] B. Byram, K. Dei, J. Tierney, and D. Dumont, ”A model and regularization scheme for ultrasonic beamforming clutter reduction,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 62, no. 11, pp. 1913–1927, 2015.

[7] A. Luchies and B. Byram, ”Deep neural networks for ultrasound beamforming,” in *Proc. of IEEE Ultrason. Symp.*, 2017.

[8] M. Gasse, F. Millioz, E. Roux, D. Garcia, H. Liebgott, D. Friboulet, ”Accelerating plane wave imaging through deep learning-based reconstruction: An experimental study,” in *Proc. of IEEE Ultrason. Symp.*, 2017.

[9] D. Perdios, A. Besson, M. Arditi, J.P. Thiran, ”A deep learning approach to ultrasound image recovery,” in *Proc. of IEEE Ultrason. Symp.*, 2017.

[10] M. Gasse, F. Millioz, E. Roux, D. Garcia, H. Liebgott, D. Friboulet, ”High-quality plane wave compounding using convolutional neural networks,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 64, no. 10, pp. 1637–39, 2017.

[11] A. C. Luchies and B. C. Byram, ”Deep neural networks for ultrasound beamforming,” *IEEE Trans. on Med. Imag.*, to be published.

[12] A. Luchies and B. Byram, ”Suppressing off-axis scattering using deep neural networks,” in *Proc. of SPIE Medical Imaging*, 2018.

[13] J. A. Jensen and N. B. Svendsen, ”Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers,” *IEEE Trans. Ultrason., Ferroelec., Freq. Contr.*, vol. 39, pp. 262–267, 1992.

[14] J. A. Jensen, ”Field: A Program for Simulating Ultrasound Systems,” in *Proc. Med. Biol. Eng. Comput.*, vol. 34, pt. 1, pp. 351–353, 1996.

[15] K. Dei and B. Byram ”The Impact of Model-Based Clutter Suppression on Cluttered, Aberrated Wavefronts,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 64, no. 10, pp. 1450–1464, 2017.

[16] B. Yang, ”A study of inverse short-time Fourier transform,” in *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2008, pp. 3541-3544.

[17] S. Ruder, ”An overview of gradient descent optimization algorithms”, arXiv:1609.04747v2, 2016.

[18] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[19] D. Kingma and J. Ba, ”Adam: A method for stochastic optimization,” in *ICLR*, 2015.

[20] X. Glorot, A. Bordes, and Y. Bengio, ”Deep sparse rectifier neural networks,” in *Proc. AISTATS*, pp. 315–323, 2011.

[21] X. Glorot and Y. Bengio, ”Understanding the difficulty of training deep feedforward neural networks,” in *Proc. AISTATS*, pp. 249–256, 2010.

[22] K. He, X. Zhang, S. Ren, and J. Sun, ”Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proc. IEEE ICCV*, pp. 1026–1034, 2015.

[23] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever,and R. Salakhutdinov, ”Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, ”Automatic differentiation in PyTorch,” *NIPS*, 2017.

[25] M. S. Patterson and F. S. Foster, ”The improvement and quantitative assessment of B-mode images produced by an annular array/cone hybrid,” *Ultrasonic Imaging*, vol. 5, no. 3, pp. 195–213, 1983.

[26] S. W. Smith, R. F. Wagner, J. M. Sandrik, and H. Lopez, ”Low contrast detectability and contrast/detail analysis in medical ultrasound,” *IEEE Trans. on Son. and Ultrason.*, vol. 30, no. 3, pp. 164–173, 1983.

[27] S. W. Smith and R. F. Wagner, ”Ultrasound speckle size and lesion signal to noise ratio: Verification of theory,” *Ultrasonic Imaging,* vol. 6, pp. 174–180, 1984.

[28] B. H. Friemel, L. N. Bohs, K. R. Nightingale, G. E. Trahey, ”Speckle decorrelation due to two-dimensional flow gradients,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 45, no. 2, pp. 317–327, 1998.

[29] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, Md. M. A. Patwary, Y. Yang, Y. Zhou, ”Deep learning scaling is predictable, empirically,” arXiv:1712.00409v1, 2017.

[30] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, ”Extracting and composing robust features with denoising autoencoders,” in *Proc. ICML*, pp. 1096–1103, 2008.

[31] P. C. Li, S. W. Flax, E. S. Ebbini, M. O'Donnell, ”Blocked element compensation in phased array imaging,” *IEEE Trans. Ultrason., Ferroelect., Freq. Control,* vol. 40, no. 4, pp. 283–292, 1993.

[32] M. Jakovljevic, G. F. Pinton, J. J. Dahl, and G. E. Trahey, ”Blocked Elements in 1-D and 2-D Arrays part I: Detection and Basic Compensation on Simulated and in vivo Targets,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 64, no. 6, pp. 910–921, 2017.

[33] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, ”The Marginal Value of Adaptive Gradient Methods in Machine Learning,” in *NIPS*, 2017.

[34] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T. P. Tang, ”On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” arXiv:1609.04836v2, 2016.

[35] K. Dei, A. Luchies, and B. Byram, ”Contrast ratio dynamic range: A new beamformer performance metric,” in *Proc. IEEE Ultrason. Symp.*, 2017.

[36] O. Rindal, A. Rodriguez-Molares, and A. Austeng, ”The dark region artifact in adaptive ultrasound beamforming,” in *Proc. IEEE Ultrason. Symp.*, 2017.