Fast ultrasonic imaging using end-to-end deep learning

Georgios Pilikos*, Lars Horchens†, Kees Joost Batenburg*‡, Tristan van Leeuwen*x and Felix Lucka*§
*Computational Imaging, Centrum Wiskunde & Informatica, Amsterdam, NL
†Applus+ E&I Technology Centre, Rotterdam, NL
‡Leiden Institute of Advanced Computer Science, Leiden University, Leiden, NL
xMathematical Institute, Utrecht University, Utrecht, NL
§Centre for Medical Image Computing, University College London, London, UK

***Abstract*—Ultrasonic imaging algorithms used in many clinical and industrial applications consist of three steps: A data pre-processing, an image formation and an image post-processing step. For efficiency, image formation often relies on an approximation of the underlying wave physics. A prominent example is the Delay-And-Sum (DAS) algorithm used in reflectivity-based ultrasonic imaging. Recently, deep neural networks (DNNs) are being used for the data pre-processing and the image post-processing steps separately. In this work, we propose a novel deep learning architecture that integrates all three steps to enable end-to-end training. We examine turning the DAS image formation method into a network layer that connects data pre-processing layers with image post-processing layers that perform segmentation. We demonstrate that this integrated approach clearly outperforms sequential approaches that are trained separately. While network training and evaluation is performed only on simulated data, we also showcase the potential of our approach on real data from a non-destructive testing scenario.**

***Index Terms*—deep learning, end-to-end training, Delay-And-Sum, fast ultrasonic imaging, approximate inversion.**

I. Introduction

Ultrasonic imaging aims at generating maps of the acoustic properties of a medium of interest. It has certain advantages over other imaging modalities such as magnetic resonance imaging (MRI) or computed tomography (CT): it uses non-ionizing radiation, it is mobile, has low operating costs and enables real-time imaging [1]. Nevertheless, the compromise in achieving fast and interactive imaging is that the resulting images require substantial human expertise for their interpretation and differentiating between materials is not trivial.

Typical workflows for 2D ultrasonic imaging with linear arrays consist of three steps: (1) data pre-processing (e.g. denoising, filtering, deconvolution), (2) image formation via beamforming and (3) image post-processing (e.g. image enhancement or segmentation). However, this three-step process introduces data/reconstruction errors which propagate due to inaccurate physics modelling or noise in the data.

Recently, there have been efforts to implement these steps with deep learning [2]. Deep neural networks (DNNs) use raw data as input and output an image [3] - [6]. The final goal is not usually to produce an image but rather it is an intermediate step before image enhancement or segmentation. Further work utilizes two decoders to obtain a beamformed image and a segmentation from one encoder using raw data [7]. Nonetheless, integrating the image formation, which approximates the underlying wave physics, within the deep learning architecture has shown to improve final results [8] [9].

This work was supported by Applus+ RTD, CWI and the Dutch Research Council (NWO 613.009.106, 639.073.506). Submission - IEEE International Ultrasonics Symposium 2020.

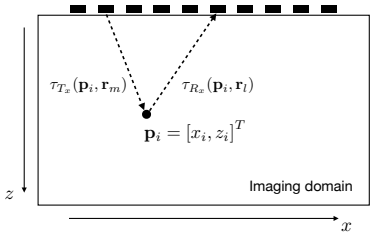


Fig. 1: A phased-array with source/receive elements ($\mathbf{r}$) depicted as black rectangles. For each image point, $\mathbf{p}_i$, and each source/receiver combination, travel times are calculated.

In this work, we propose to integrate all three steps together to enable end-to-end training. To achieve this, we propose a novel architecture that utilizes a fast ultrasonic imaging operator, the Delay-And-Sum (DAS). We examine turning the DAS image formation into a network layer that connects data pre-processing and image post-processing DNNs. Using this, we propose an end-to-end training strategy to obtain improved results. In section 2, we describe the ultrasonic data acquisition and the DAS image formation. Then, in section 3, we introduce our proposed end-to-end deep learning approach and in section 4, we demonstrate that our integrated approach outperforms sequential approaches which are trained independently using simulated data. Finally, we apply this to real data showcasing its potential to a non-destructive testing scenario.

II. Ultrasonic imaging

We examine 2D data acquisition with a linear array as shown in Figure 1. An element is used as a source, $\mathbf{r}_m$, and transmits a pulsed ultrasonic wave into the medium of interest. All receivers capture the resulting wave field, which contains information about the wave-matter interactions, e.g. via reflections from interfaces with different acoustic properties. The data acquisition continues with the next element as a source and so on until all elements have acted as sources [10], [11].

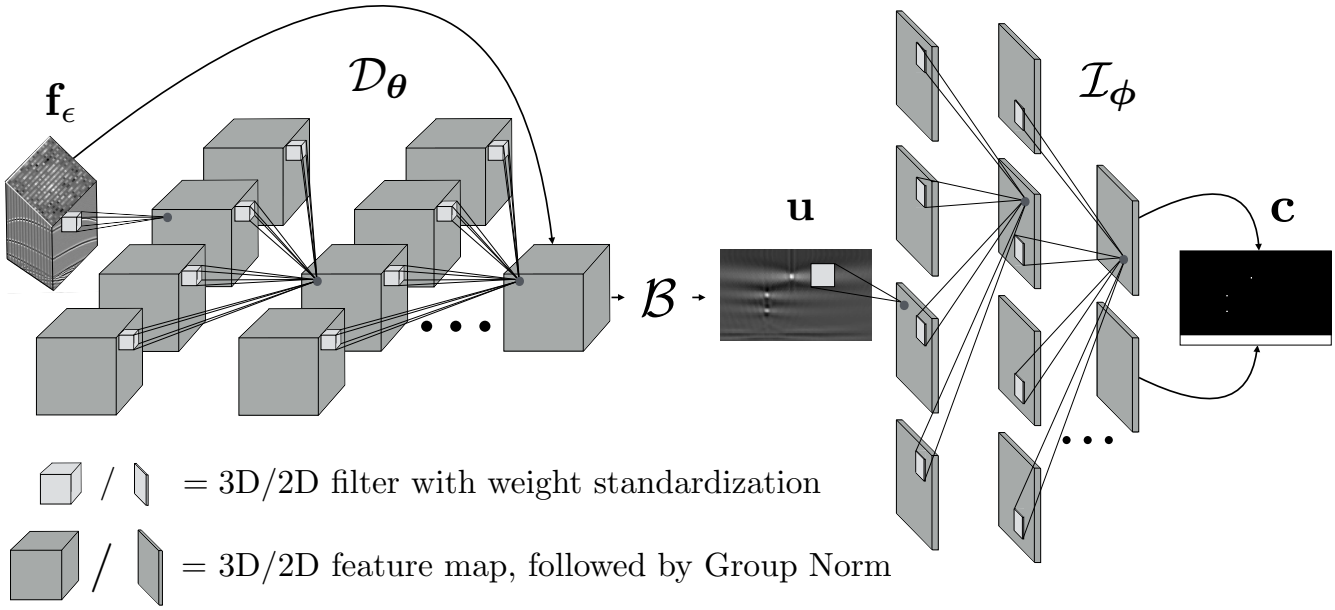


Fig. 2: A 3D DCNN, $\mathcal{D}_{\boldsymbol{\theta}}$, is used for data pre-processing and the DAS operator, $\mathcal{B}$, incorporates the image formation into the whole network. A 2D DCNN, $\mathcal{I}_{\boldsymbol{\phi}}$, post-processes the intermediate image and produces the final result. Feature maps are followed by Group Norm and ReLU (tanh is used at the last layer of $\mathcal{D}_{\boldsymbol{\theta}}$). Only one filter at one location per layer is shown.

This is called Full Matrix Capture (FMC) and leads to a data volume, $\mathbf{f} \in \mathbb{R}^{n_t \times n_s \times n_r}$, where n_t, n_s and n_r are the number of time samples, sources and receivers respectively. The aim is to obtain an image, $\mathbf{u} \in \mathbb{R}^{n_x \times n_z}$, where n_x and n_z are the number of pixels in the horizontal and vertical direction.

Delay-And-Sum image formation

Delay-And-Sum (DAS) relies on an approximation of the underlying wave physics. It calculates travel times, $\tau(\mathbf{p}_i, \mathbf{r}_m, \mathbf{r}_l)$, between each source, $\mathbf{r}_m$, each image point, $\mathbf{p}_i = [x_i, z_i]^T$ and each receiver, $\mathbf{r}_l$, assuming a uniform speed of sound in the material, s. This is calculated by,

$$\tau(\mathbf{p}_i, \mathbf{r}_m, \mathbf{r}_l) = \frac{\|\mathbf{r}_m - \mathbf{p}_i\|_2}{s} + \frac{\|\mathbf{r}_l - \mathbf{p}_i\|_2}{s}, \tag{1}$$

and depicted in Figure 1. Each travel time is converted into an index using the sampling frequency which is used to locate a sample. This time-shift operation corresponds to the *delay* part. The amplitude is extracted at that time-shifted location, and the process is repeated for all travel times corresponding to an image point. Finally, it *sums* all amplitudes giving image amplitude, u_i, for image point, $\mathbf{p}_i$. This can be written as,

$$u_i = \sum_{m=0}^{n_s} \sum_{l=0}^{n_r} f(\tau(\mathbf{p}_i, \mathbf{r}_m, \mathbf{r}_l), m, l), \tag{2}$$

and repeated for all image points to form an image. We can write it as a linear operator, $\mathcal{B} : \mathbb{R}^{n_t \times n_s \times n_r} \rightarrow \mathbb{R}^{n_x \times n_z}$, and referred as *DAS operator*. The whole process is written as,

$$\mathbf{u} = \mathcal{B}\mathbf{f}. \tag{3}$$

Note that the true mapping from acoustic properties to data is non-linear and includes a lot of different, complicated wave-matter interactions. On the other hand, the DAS algorithm corresponds to a linear back-projection-type operator that tries to form an approximate, qualitative image of acoustic property variations in space. Due to this approximation, it is usually preceded by data pre-processing and followed by image post-processing. These individual steps are increasingly being replaced by deep convolutional neural networks (DCNNs) [2].

III. End-to-end deep learning

DCNNs are parameterized non-linear mappings optimized for a given loss function. In this work, we propose a novel architecture, as shown in Figure 2. The architecture involves a 3D DCNN, which is a data-to-data mapping $\mathcal{D}_{\boldsymbol{\theta}}$, acting on the data volume. Then, we incorporate the DAS operator, $\mathcal{B}$, into the network, by implementing a layer that applies the DAS algorithm on the data. For this, we also need to allow backpropagation of errors during training by deriving and implementing its adjoint action. Finally, the intermediate image formed is processed by a 2D DCNN, which is an image-to-image mapping $\mathcal{I}_{\boldsymbol{\phi}}$, to obtain a segmented image. This enables end-to-end training of DCNN parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ simultaneously. Both DCNNs have 4 layers with 4 filters, each with $5 \times 5 \times 5$ and 5×5 dimensions. Weight Standardization [12] and Group Normalization [13] is used per layer to help training stability since we use one training sample per mini-batch. Furthermore, skip connections in $\mathcal{D}_{\boldsymbol{\theta}}$ enable better information flow and reduce training time [14].

A. Training strategies

We will examine two sequential training strategies and introduce our proposed end-to-end approach. To facilitate discussion, we define $\mathbf{c}^{(i)}$ as the ground truth segmentation, $\mathbf{f}^{(i)}$ as clean simulated data, $\mathbf{f}_\epsilon^{(i)}$ as noisy, undersampled data and $\mathbf{u}^{(i)}$

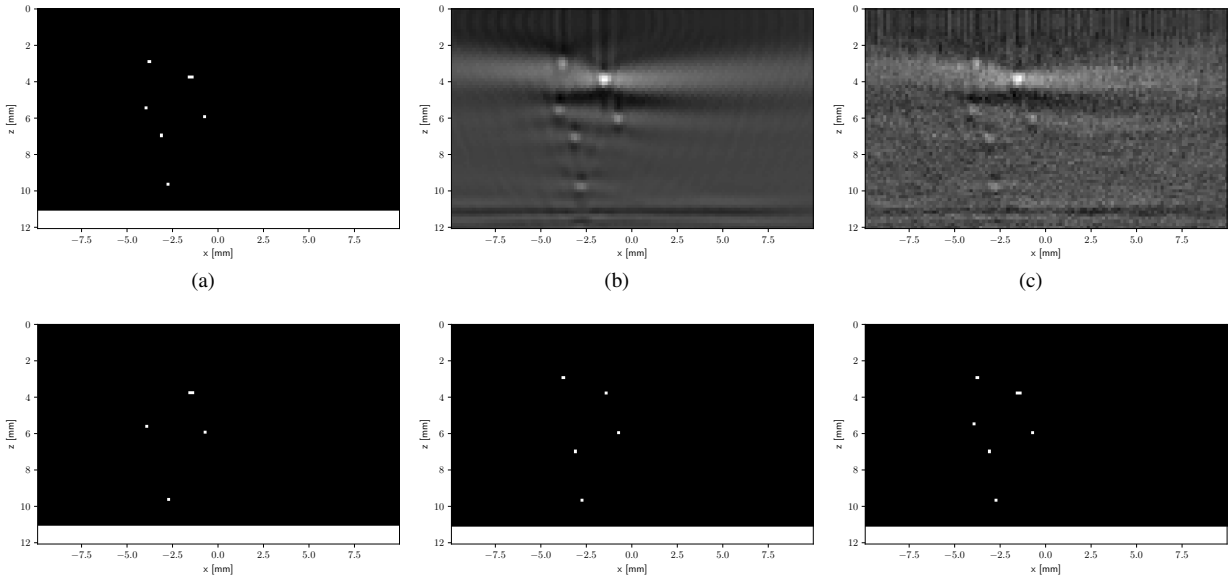


Fig. 3: (a) Target segmentation (white: air, black: carbon steel), (b) DAS image from fully-sampled data, (c) DAS image from noisy and under-sampled data, (d)-(f) segmentation result of 1st/2nd/3rd training strategy from noisy and under-sampled data.

as the DAS image from $\mathbf{f}^{(i)}$ using equation 3. The superscript i represents the i-th training sample from a collection of training data, $\{\mathbf{c}^{(i)}, \mathbf{f}^{(i)}, \mathbf{f}_\epsilon^{(i)}, \mathbf{u}^{(i)}\}_{i=1}^N$. All training strategies use the same loss function for image post-processing, namely the cross entropy loss referred to as $\mathcal{H}$ hereafter. The strategies are:

1st training strategy: Data pre-processing DCNN, $\mathcal{D}_{\boldsymbol{\theta}}$, is trained and fixed. The DAS operator, $\mathcal{B}$, is applied to pre-processed data to form images. Then, image post-processing DCNN, $\mathcal{I}_{\boldsymbol{\phi}}$, is trained using these images. That is,

1) train $\hat{\boldsymbol{\theta}} := \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^N \|\mathbf{f}^{(i)} - \mathcal{D}_{\boldsymbol{\theta}}(\mathbf{f}_\epsilon^{(i)})\|_2^2$
2) compute $\hat{\mathbf{u}}^{(i)} := \mathcal{B}\mathcal{D}_{\hat{\boldsymbol{\theta}}}(\mathbf{f}_\epsilon^{(i)})$
3) train $\hat{\boldsymbol{\phi}} := \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \sum_{i=1}^N \mathcal{H}(\mathbf{c}^{(i)}, \mathcal{I}_{\boldsymbol{\phi}}(\hat{\mathbf{u}}^{(i)}))$

2nd training strategy: Data pre-processing DCNN, $\mathcal{D}_{\boldsymbol{\theta}}$, and DAS operator, $\mathcal{B}$, are trained together. Then, image post-processing DCNN, $\mathcal{I}_{\boldsymbol{\phi}}$, is trained. That is,

1) train $\hat{\boldsymbol{\theta}} := \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^N \|\mathbf{u}^{(i)} - \mathcal{B}\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{f}_\epsilon^{(i)})\|_2^2$
2) compute $\hat{\mathbf{u}}^{(i)} := \mathcal{B}\mathcal{D}_{\hat{\boldsymbol{\theta}}}(\mathbf{f}_\epsilon^{(i)})$
3) train $\hat{\boldsymbol{\phi}} := \underset{\boldsymbol{\phi}}{\operatorname{argmin}} \sum_{i=1}^N \mathcal{H}(\mathbf{c}^{(i)}, \mathcal{I}_{\boldsymbol{\phi}}(\hat{\mathbf{u}}^{(i)}))$

3rd training strategy: All three steps are combined and trained together in an end-to-end way as proposed. That is,

1) train $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}}) := \underset{(\boldsymbol{\phi},\boldsymbol{\theta})}{\operatorname{argmin}} \sum_{i=1}^N \mathcal{H}(\mathbf{c}^{(i)}, \mathcal{I}_{\boldsymbol{\phi}}(\mathcal{B}\mathcal{D}_{\boldsymbol{\theta}}(\mathbf{f}_\epsilon^{(i)})))$

For the 2nd and 3rd strategies, we initialize the DCNNs with the parameters learnt by 1st and 2nd strategies respectively.

IV. Experiments

To evaluate our proposed approach, we use an ultrasonic-based non-destructive inspection of pipelines for defects.

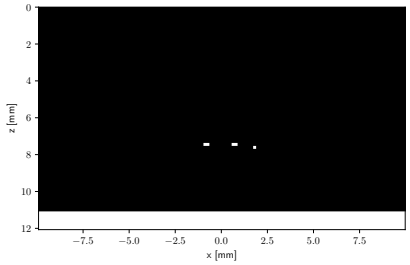
A. Simulated data

The data domain was set to $64 \times 64 \times 1020$ with 64 elements, 1020 time samples and sampling frequency of 50MHz. The image domain was set to 72×354 pixels with defects randomly located around the middle of the domain. This was then cropped to 72×118, as used in the real data acquired. As a proof of concept, we set the number of materials to 2. The segmented image consists of 0 or 1 which corresponds to the speed of sound of each material.

Figure 3(a) includes an example of a speed of sound map. The pipeline was modelled as carbon steel ($s = 5920$m/s) and the defects and pipe wall as air ($s = 343$m/s). We simulated ultrasonic data with k-Wave [15] and used them as input for training. The respective speed of sound maps were used as targets. Each simulation took approximately 5 minutes to run on an NVIDIA Geforce GTX 970. We limited the generation to only 230 scenarios (training data: 180, test data: 50) where we randomly varied the number and location of defects in a pipeline. To increase difficulty, we added noise and under-sampled sources by a factor of two. An example of a DAS image using clean, fully-sampled data is shown in Figure 3(b). The defects are correctly located but there are side lobes present due to the limited spatial coverage of the linear array. Figure 3(c) includes the DAS image from noisy, under-sampled data. In this case, it is more challenging to localize the defects in the ultrasonic image.

We use the noisy and under-sampled data to evaluate the three training strategies introduced in the previous section. All strategies are implemented in PyTorch, and DCNN parameters are optimized using the Adam optimization [16] with a learning rate of 10^{-3}. The average cross entropy (lower is better)

(a)



(b)

Fig. 4: (a) Picture of a carbon steel block, (b) segmentation result of proposed end-to-end deep learning approach.

of each strategy on the test set is: 3.4×10^{-3}, 6.7×10^{-4} and 1.2×10^{-4} each. A visual comparison is given in Figure 3(d), 3(e) and 3(f) where the segmented images obtained by each strategy are shown. These results demonstrate that the proposed end-to-end integration of the DAS operator with both data pre-processing and image post-processing steps leads to a substantial improvement of the final segmentation result.

B. Real data

To further validate our proposed approach, we acquired real ultrasonic data using a carbon steel block with three holes. A picture can be seen in Figure 4(a). We used the data acquired (half of the sources) as input and estimated a segmented image. The spatial extent of the defects is underestimated since we did not take into account the temporal impulse response of the receivers during training data simulation. Nevertheless, we obtain an accurate localization and separation of the defects as seen in Figure 4(b). Our end-to-end deep learning approach was trained only on simulated data but it was able to transfer the learnt representations to more challenging real data.

V. Conclusions

Deep learning can be integrated into existing ultrasonic imaging workflows and replace traditional data pre-processing and image post-processing steps with success. Nevertheless, there are various architectures and strategies for training deep neural networks. In this work, we proposed an end-to-end approach that integrates the image formation into the network architecture. This results in a single network that maps raw data to desired imaging result. We demonstrated this concept for the DAS image formation and for segmentation as an image post-processing task. To increase difficulty, we sub-sampled the noisy data by half, which could be used to speed up the data acquisition in real-world applications. Experiments have shown that end-to-end training produces better segmented images as opposed to training for each task separately. Even though the final cost function is the same, sub-optimal results are obtained when training steps sequentially. This is because we fix the parameters of the data pre-processing network and only optimize the parameters of the image post-processing network. On the other hand, end-to-end training is more flexible since it optimizes the parameters of both networks simultaneously. It is initialized with the learnt network parameters of the sequential approach and can only improve upon those. Furthermore, training was performed only on simulated data, but the proposed approach was successful on real data illustrating the potential of deep learning to learn from physics simulations to solve real-world ultrasonic inverse problems.

References

[1] M. Tanter and M. Fink, "Ultrafast imaging in biomedical ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 61, no. 1, pp. 102–119, 2014.

[2] R. J. G. van Sloun, R. Cohen, and Y. C. Eldar, "Deep learning in ultrasound imaging," *Proceedings of the IEEE*, vol. 108, no. 1, pp. 11–29, 2020.

[3] S. Khan, J. Huh, and J. C. Ye, "Adaptive and compressive beamforming using deep learning for medical ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020.

[4] B. Luijten, R. Cohen, F. J. de Bruijn, H. A. W. Schmeitz, M. Mischi, Y. C. Eldar, and R. J. G. van Sloun, "Deep learning for fast adaptive beamforming," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1333–1337.

[5] D. Perdios, M. Vonlanthen, F. Martinez, M. Arditi, and J. Thiran, "Deep learning based ultrasound image reconstruction method: A time coherence study," in *2019 IEEE International Ultrasonics Symposium (IUS)*, 2019, pp. 448–451.

[6] W. Simson, R. Göbl, M. Paschali, M. Krönke, K. Scheidhauer, W. Weber, and N. Navab, "End-to-end learning-based ultrasound reconstruction," *CoRR*, vol. abs/1904.04696, 2019. [Online]. Available: http://arxiv.org/abs/1904.04696

[7] A. A. Nair, K. N. Washington, T. D. Tran, A. Reiter, and M. A. L. Bell, "Deep learning to obtain simultaneous image and segmentation outputs from a single input of raw ultrasound channel data," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020.

[8] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," 2020.

[9] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, "Solving inverse problems using data-driven models," *Acta Numerica*, vol. 28, pp. 1–174, 2019.

[10] C. Holmes, B. W. Drinkwater, and P. D. Wilcox, "Post-processing of the full matrix of ultrasonic transmit receive array data for non-destructive evaluation," *NDT & E International*, vol. 38, no. 8, pp. 701 – 711, 2005.

[11] N. Portzgen, D. Gisolf, and G. Blacquiere, "Inverse wave field extrapolation: a different NDI approach to imaging defects," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 54, no. 1, pp. 118–127, 2007.

[12] S. Qiao, H. Wang, C. Liu, W. Shen, and A. L. Yuille, "Weight standardization," *CoRR*, vol. abs/1903.10520, 2019. [Online]. Available: http://arxiv.org/abs/1903.10520

[13] Y. Wu and K. He, "Group normalization," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241.

[15] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of Biomedical Optics*, vol. 15, no. 2, pp. 1 – 12, 2010. [Online]. Available: https://doi.org/10.1117/1.3360308

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, 2015.