# A Hybrid Deep Learning Pipeline for Improved Ultrasound Localization Microscopy

Tristan S.W. Stevens[⋆], Elizabeth B. Herbst[‡], Ben Luijten[⋆]
Boudewine W. Ossenkoppele[§,⋆], Thierry J. Voskuil[⋆], Shiying Wang[‡], Jihwan Youn[⋆]
Claudia Errico[‡], Massimo Mischi[⋆], Nicola Pezzotti[⋆†], Ruud J.G. van Sloun[⋆†]

[⋆] Dept. of Electrical Engineering, Eindhoven University of Technology, The Netherlands
[§] Dept. of Imaging Physics, Delft University of Technology, The Netherlands
[†] Philips Research, Eindhoven, The Netherlands, [‡] Philips Research, North America

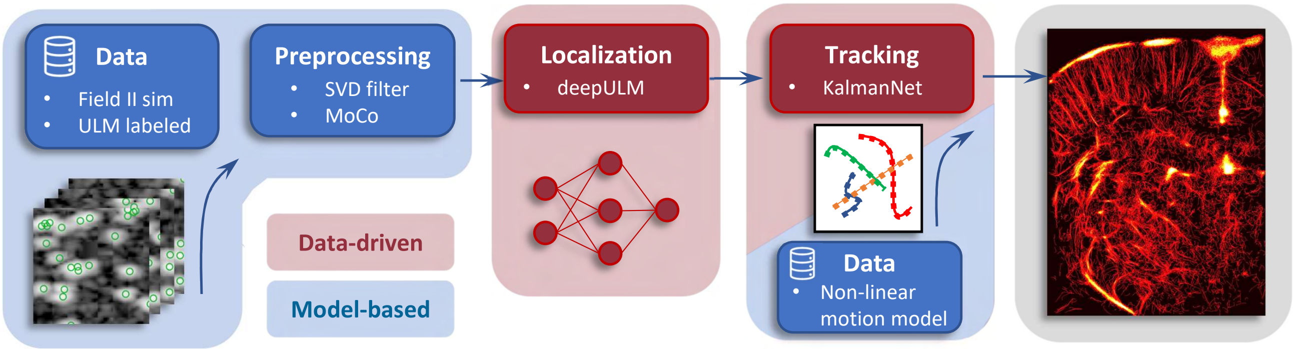


Fig. 1: Hybrid ultrasound localization microscopy (ULM) pipeline with model-based (blue) and data-driven (red) elements.

***Abstract*—The image quality of ultrasound localization microscopy (ULM) images is driven by the ability to accurately detect and track the location of microbubbles (MBs) in vascular networks. This task becomes increasingly challenging in imaging environments with high MB concentrations and low signal-to-noise ratios, making it difficult to differentiate and localize individual MBs. Recent developments in deep learning (DL) have demonstrated significant improvements over conventional methods but depend on vast amounts of realistic training data with the corresponding ground truth labels, which are difficult to obtain. The alternative, simulated data, in turn, poses challenges in generalizability of the method. In this work, we present a hybrid pipeline for ULM that comprises data generation, localization, and tracking. It combines the current state-of-the-art, utilizing both conventional and DL techniques. We show that using this approach, we can create high-quality velocity maps while being able to generalize well across different domains.**

## I. Introduction

Ultrasound localization microscopy (ULM) is an exciting and upcoming field of research in biomedical ultrasound (US). Traditionally, ULM image reconstruction is divided into 4 steps: 1) filtering and motion compensation, 2) microbubble (MB) localization, 3) MB tracking and 4) image formation [1], [2]. Most of the time, these algorithms rely on careful parameter tuning or approximations of the underlying measurement model, limiting its imaging potential. More recently, deep learning has proven itself as an effective tool for ULM, especially for pre-processing and the localization of MBs [3], [4]. However, several challenges remain. High bubble concentrations hamper the localization of MBs due to overlapping PSFs. On the contrary, distinguishing individual MBs is easier at low concentrations, but a longer acquisition time is needed to produce sufficient image quality. In turn, such an acquisition may be subject to stronger motion artifacts in *in-vivo* scenarios.

In this work, our aim is to tackle these issues by employing robust conventional processing, together with state-of-the-art deep learning techniques. Traditional image processing methods are used to estimate the number of MBs per frame and to estimate the PSF properties. Subsequently, we perform sub-pixel localization of the MB centers through a convolutional neural network based on the work by van Sloun *et al.* [4], which is trained on a mixture of simulation and *in-vivo* data. Finally, we use a novel model-based MB tracking method based on KalmanNet [5], a hybrid network (model-based *and* data-driven) that combines classical Kalman filtering with a recurrent neural network (RNN). An overview of the complete hybrid pipeline is shown in Fig. 1.

## II. ULTRA-SR challenge

The Ultrasound Localization and TRacking Algorithms for Super Resolution (ULTRA-SR) challenge is a challenge organized by the 2022 International Ultrasonics Symposium (IUS). The aim of the challenge is to advance the field of ULM imaging. The challenge data comprises four datasets, two synthetically generated (linear and phased array) and two *in-vivo* (rat brain and lymph node) B-mode US sequences. There are three categories in the competition including 1) localization – synthetic data 2) localization and tracking – synthetic data, and 3) localization and tracking - *in-vivo* data.

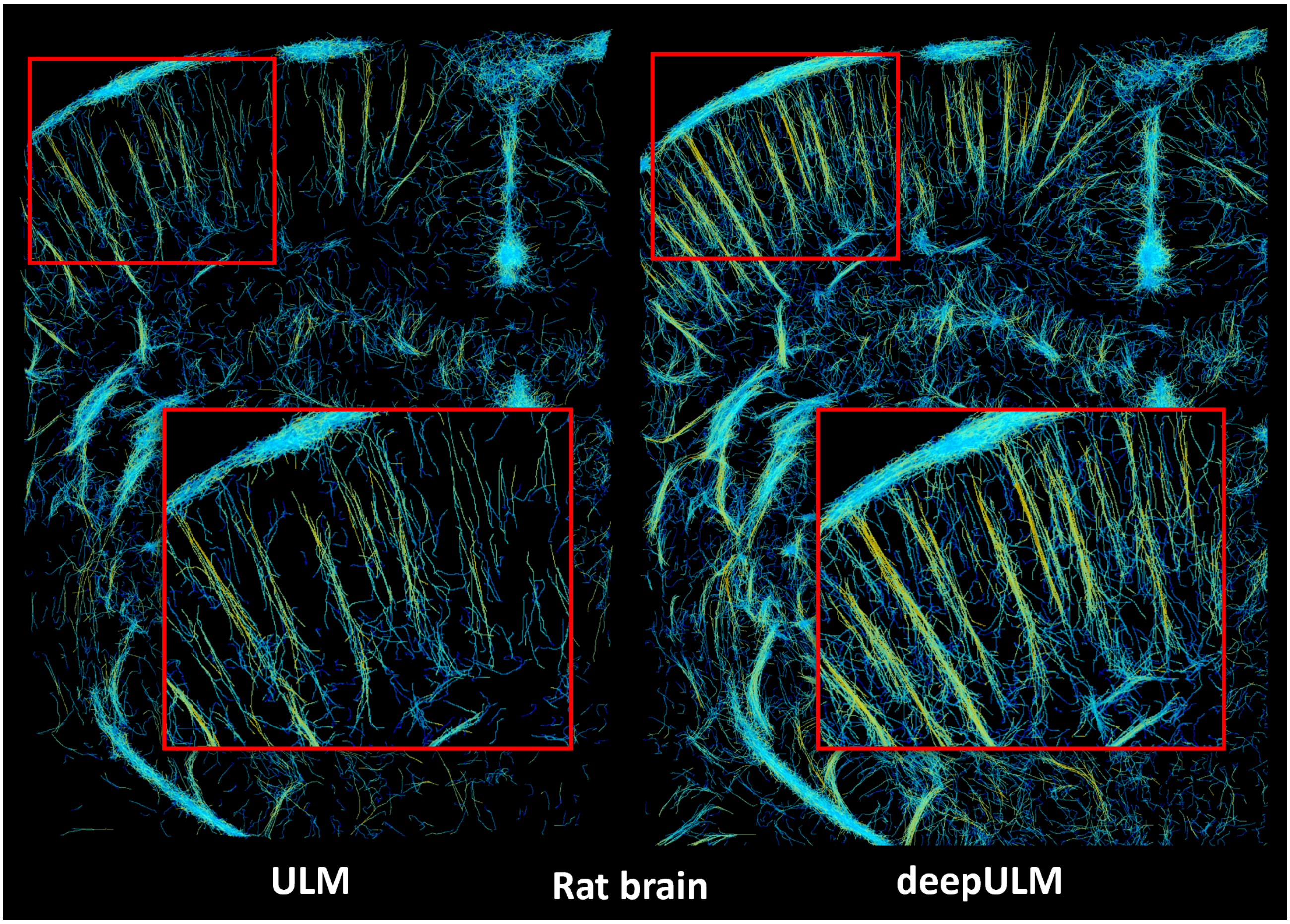


Fig. 2: Results on the *in-vivo* rat brain data of the challenge. Comparing both ULM and deepULM images.

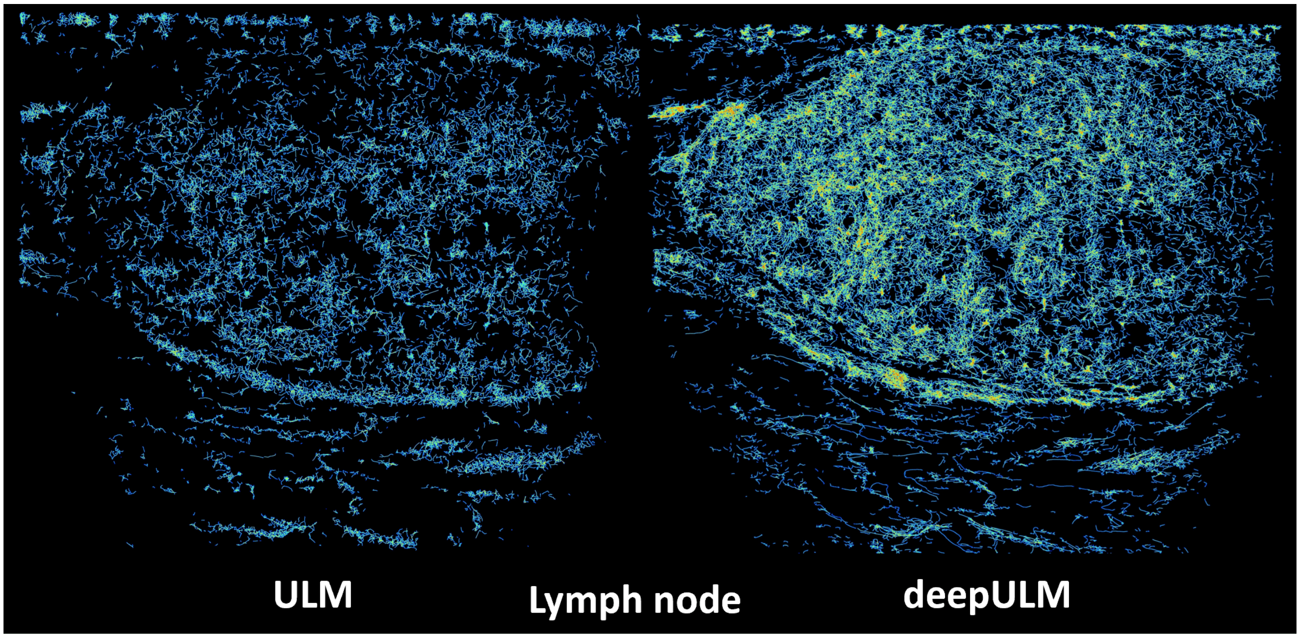


Fig. 3: Results on the *in-vivo* lymph node data of the challenge. Comparing both ULM and deepULM images.

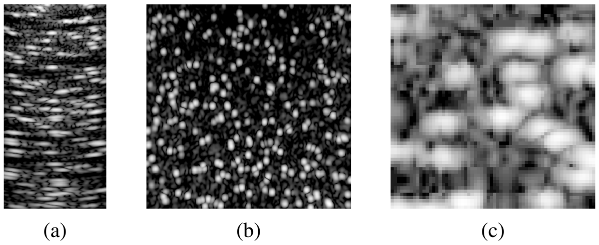


Fig. 4: Examples of the simulated images used for training: (a) phased array synthetic data, (b) linear array synthetic data, and (c) lymph node data.

## III. Methods

### A. Preprocessing

Before DL-based methods are applied to the challenge data, conventional image processing methods are used to provide baseline ULM images. For the *in vivo* lymph node data set, we apply clutter filtering using a slow-time highpass FIR filter. This clutter-filtered data set is used as input to the DL pipeline. For all other data sets, we do not apply preprocessing. After preprocessing, ULM is performed using performance assessment localization algorithm (PALA) based methods [2], [6]. For each data set, the PALA method parameters are tuned to account for the estimated size of the MB point spread function (PSF) and the estimated number of MB in a frame. Final sub-pixel localization of the MB center is performed using a weighted average method.

The results from this conventional ULM pipeline are used for two purposes. First, the localization data is used as a quasi-ground truth data set for neural network training. Second, the output MB density maps from the conventional ULM pipeline are used as a basis of comparison against DL-based density maps (Fig. 2 and Fig. 3). In the following paragraphs we will detail on the subsequent processing steps.

### B. Localization

*1) Data generation:* For the synthetic and lymph node datasets, training data are simulated using the Field II ultrasound simulation [7]. In the ideal scenario, we would like to train the networks with the provided data. However, it is challenging to obtain the ground truth, that is, true MB positions, especially when the MBs are placed closer than the resolution limit of ultrasound in one image frame. For the synthetic phased array dataset, the peak intensity of an isolated PSF does not correspond to the MB position when the MB is away from the center laterally. This happens because the array size is small and at the same time the imaging region is deep. Furthermore, some imaging regions are not fully covered by transmitted plane waves. Additionally, the number of frames for the synthetic datasets is not large enough to be used for training.

To simulate MB images, point scatterers are placed randomly in the region of interest, and ultrasound channel data are simulated using plane waves. Next, delay-and-sum beamforming with dynamic apodization and compounding is performed to form the MB ultrasound image. Most imaging parameters such as plane wave angles, apodization window, and the F-number are chosen empirically, as the true parameters for the synthetic data are unknown. For the synthetic training sets, clutter images are also simulated separately in the same way but using 10 to 20 times more point scatterers than the MB images. A training example is then created by randomly selecting one MB image and one clutter image and applying envelope detection to the summation of them. For the lymph node training set, the clutter images are not simulated but extracted from the provided measurements, as they included compression artifacts that cannot easily be simulated. For the lymph node training set, the clutter images are already envelope detected, so we cannot simply sum an MB image with a clutter image. For a smooth transition between the MB and clutter images, a weighted summation of an envelope detected MB image and a clutter image is used:

$$I_{tr} = I_{MB} + (\max(I_{MB}) - I_{MB}) \times I_{clutter}, \tag{1}$$

where $I_{tr}$ is the training data, $I_{MB}$ is the MB image, and $I_{clutter}$ is the clutter image. Examples of simulated training data are shown in Fig. 4.

For the rat brain dataset, the MB density is low, and the number of frames is large enough for training. Therefore, we directly employ the provided data for training with the MB positions estimated by a conventional method.

*2) Network architecture and training:* For MB localization we employ the U-net architecture presented in [4], and upscale the input by a factor of 10. Training is facilitated through a mean-squared-error (MSE) between the predicted locations and the ground truth targets on input patches of $64 \times 64$ pixels. Furthermore, we apply a small Gaussian blur to the target labels to relax the MSE objective and favor detection accuracy over localization error. As a result, we are able to train the localization network in a regression style, such that detections that are a few pixels off still contribute to the loss. Finally, we perform a hyperparameter search to optimize the number of layers and kernel sizes in the network. The predictions of the best performing models are subsequently ensembled to remove outliers through majority voting.

### C. Tracking

Our proposed tracking framework combines the advanced motion estimation of KalmanNet [5] with the conventional data association of the Hungarian Algorithm (HA) [8]. KalmanNet is used for the motion estimation of the MBs and predicts the positions of the MBs in the next frame. Subsequently, HA is used to solve the linear sum assignment problem with a pairwise distance matrix as cost.

The process of KalmanNet is similar to classic Kalman filtering; it also has a prediction and an update step to estimate states. However, the Kalman gain computation is learned from data through an RNN. Furthermore, KalmanNet does not explicitly track the covariance matrix.

*1) Data generation:* KalmanNet is trained on a dataset containing tracks that were generated with a nonlinear motion model (NMM) in state space format, as done in [9]. The state vector $\mathbf{x}_t$ of an MB for frame $t$ is given as $\mathbf{x}_t = (x_t, z_t, v_t, \varphi_t, \omega_t)^\top$, where $x_t$ and $z_t$ are the lateral and axial coordinates, respectively. The velocity is denoted by $v_t$ and the direction is determined by the angle $\varphi_t$ and the turning rate $\omega_t$. This motion model takes into account the curved physiology of vessels by including the turn rate $\omega_t$. A more detailed description on the track generation is given in [9]. In total, 25000 tracks of length 100 were generated at a frame rate of 100 Hz with a maximum flow velocity of 15 mm/s.

*2) KalmanNet training:* In KalmanNet, the Kalman gain computation is performed by an RNN. The network has a fully connected (FC) layer, which is followed by a Gated Recurrent Unit (GRU) layer. The GRU layer is succeeded by two other FC layers. The last FC layer transforms the output to the number of features in the Kalman gain. An L2 regularized Mean Squared Error (MSE) loss function is used. The loss function is based on the error in the predicted and ground truth state and is given as follows,

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} \left(\hat{\mathbf{x}}_t - \mathbf{x}_t\right)^2 + \lambda \left\lVert w \right\rVert_2^2, \tag{2}$$

where $T$ is the sequence length of the tracks, $\lambda$ is the regularization parameter, $w$ are the weights of the network, and $\hat{\mathbf{x}}_t$ and $\mathbf{x}_t$ are the estimated and ground truth state, respectively. The loss function is minimized with the Adam optimizer. The hyperparameters were optimized through a random search.

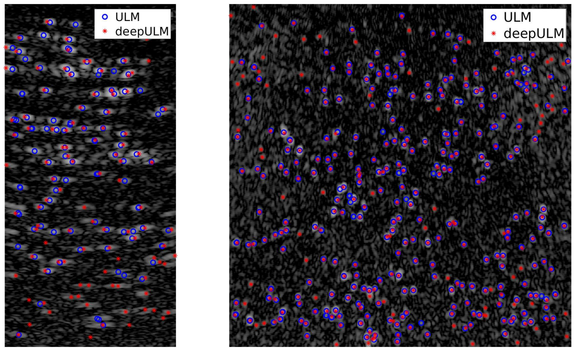
*3) Tracker:* The KalmanNet-based tracker consists of three steps that are executed for every frame, namely prediction, data association, and update. In the prediction step, KalmanNet estimates the next MB position on its corresponding track. Consequently, in the data association step a cost matrix, containing the Euclidean distances from predicted MB positions to localized MB positions, is constructed and used by the HA to solve the assignment problem. After applying the HA, the assignments are checked against a distance threshold to avoid physical implausible displacements. Lastly, in the update step, the assigned observations are used to update the state and Kalman gain of the corresponding tracks.

## IV. Results and Conclusion

The synthetic datasets are not labeled, since they are part of the test set of the challenge. Similarly, the *in-vivo* datasets are also not labeled. However, the performance can still be qualitatively evaluated. We visually compare our hybrid method with a conventional ULM method [6].

### A. Synthetic data

*1) GE M5Sc-D Phased array:* MB localizations for the low frequency phased array data are shown in Fig. 4a. The deepULM localization network is more sensitive in the deeper region. However, deepULM sometimes detects two MBs from one isolated MB.



(a)

(b)

Fig. 5: Results on the synthetic datasets: (a) phased array with a low frequency and (b) linear array with a high frequency.

*2) Verasonics L11-4v Linear array:* MBs localizations for the high frequency linear array data are shown in Fig. 4b. ULM and deepULM show similar results, since the high frequency data results in smaller PSF and less overlapping between MBs. Still, deepULM can resolve closely spaced MBs where ULM does not in some cases.

### B. In-vivo data

The 2D velocity (magnitude) ULM images for the rat brain and lymph node *in-vivo* data are shown in Fig. 2 and Fig. 3, respectively. Our method shows increased sensitivity with respect to the conventional method, and cleaner tracks overall. It is possible to achieve more sensitivity with the conventional method, however, we found that the resulting tracks were noisier, possibly due to an increase in false detections.

## References

[1] K. Christensen-Jeffries, O. Couture, P. A. Dayton, Y. C. Eldar, K. Hynynen, F. Kiessling, M. O'Reilly, G. F. Pinton, G. Schmitz, M.-X. Tang, M. Tanter, and R. J. van Sloun, "Super-resolution ultrasound imaging," *Ultrasound Med. Biol.*, vol. 46, no. 4, pp. 865–891, 2020. 1

[2] O. Couture, V. Hingot, B. Heiles, P. Muleki-Seya, and M. Tanter, "Ultrasound localization microscopy and super-resolution: A state of the art," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 65, no. 8, pp. 1304–1320, 2018. 1, 3

[3] K. G. Brown, D. Ghosh, and K. Hoyt, "Deep learning of spatiotemporal filtering for fast super-resolution ultrasound imaging," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 67, no. 9, pp. 1820–1829, 2020. 1

[4] R. J. G. van Sloun, O. Solomon, M. Bruce, Z. Z. Khaing, H. Wijkstra, Y. C. Eldar, and M. Mischi, "Super-resolution ultrasound localization microscopy through deep learning," *IEEE Trans. Med. Imaging*, vol. 40, no. 3, pp. 829–839, 2021. 1, 3

[5] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. Van Sloun, and Y. C. Eldar, "Kalmannet: Neural network aided kalman filtering for partially known dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022. 1, 3

[6] B. Heiles, A. Chavignon, V. Hingot, P. Lopez, E. Teston, and O. Couture, "Performance benchmarking of microbubble-localization algorithms for ultrasound localization microscopy," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 6, no. 5, pp. 605–616, 2022. 3, 4

[7] J. A. Jensen, "Field: A program for simulating ultrasound systems," *Med. Biol. Eng. Comput.*, vol. 34, pp. 351–353, 1996. 3

[8] H. W. Kuhn, "The hungarian method for the assignment problem," *Nav. Res. Logist. Q.*, vol. 2, no. 1-2, pp. 83–97, 1955. 3

[9] M. Piepenbrock, S. Dencks, and G. Schmitz, "Microbubble tracking with a nonlinear motion model," *IEEE International Ultrasonics Symposium, IUS*, vol. 2020-September, no. 1, pp. 2020–2023, 2020. 4