

PAPER • OPEN ACCESS

Offline and online LSTM networks for respiratory motion prediction in MR-guided radiotherapy

To cite this article: Elia Lombardo *et al* 2022 *Phys. Med. Biol.* **67** 095006

View the [article online](#) for updates and enhancements.

You may also like

- [Spatio-temporal warping for myoelectric control: an offline, feasibility study](#)
Milad Jabbari, Rami Khushaba and Kianoush Nazarpour
- [High-precision time delay estimation of narrowband radio signal by PHAT-LSTM](#)
Shunyu Yao, Qiao Meng, Congyan Chen et al.
- [Long exposure convolutional memory network for accurate estimation of finger kinematics from surface electromyographic signals](#)
Weiyu Guo, Chenfei Ma, Zheng Wang et al.

VERIQA
RT MonteCarlo 3D
Plan selected. Plan verified.
In less than 3 minutes.

Automated. Independent. Web-Based.

PTW THE DOSIMETRY COMPANY

Explore the benefits of streamlined patient QA



PAPER

Offline and online LSTM networks for respiratory motion prediction in MR-guided radiotherapy

OPEN ACCESS

RECEIVED

13 December 2021

REVISED

1 March 2022

ACCEPTED FOR PUBLICATION

24 March 2022

PUBLISHED

19 April 2022

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Elia Lombardo¹ , Moritz Rabe¹ , Yuqing Xiong¹, Lukas Nierer¹, Davide Cusumano² , Lorenzo Placidi², Luca Boldrini², Stefanie Corradini¹, Maximilian Niyazi¹, Claus Belka^{1,3}, Marco Riboldi⁴ , Christopher Kurz¹ and Guillaume Landry¹

¹ Department of Radiation Oncology, University Hospital, LMU Munich, Munich, D-81377, Germany

² Fondazione Policlinico Universitario Agostino Gemelli IRCCS, Rome, I-00168, Italy

³ German Cancer Consortium (DKTK), Partner Site Munich, Munich, D-81377, Germany

⁴ Department of Medical Physics, Faculty of Physics, Ludwig-Maximilians-Universität München, Garching, D-85748, Germany

E-mail: guillaume.landry@med.uni-muenchen.de

Keywords: AI, long short-term memory networks, time series prediction, respiratory motion, MR-linac, MRI-guidance

Supplementary material for this article is available [online](#)

Abstract

Objective. Gated beam delivery is the current clinical practice for respiratory motion compensation in MR-guided radiotherapy, and further research is ongoing to implement tracking. To manage intra-fractional motion using multileaf collimator tracking the total system latency needs to be accounted for in real-time. In this study, long short-term memory (LSTM) networks were optimized for the prediction of superior–inferior tumor centroid positions extracted from clinically acquired 2D cine MRIs. **Approach.** We used 88 patients treated at the University Hospital of the LMU Munich for training and validation (70 patients, 13.1 h), and for testing (18 patients, 3.0 h). Three patients treated at Fondazione Policlinico Universitario Agostino Gemelli were used as a second testing set (1.5 h). The performance of the LSTMs in terms of root mean square error (RMSE) was compared to baseline linear regression (LR) models for forecasted time spans of 250 ms, 500 ms and 750 ms. Both the LSTM and the LR were trained with offline (*offline* LSTM and *offline* LR) and online schemes (*offline+online* LSTM and *online* LR), the latter to allow for continuous adaptation to recent respiratory patterns. **Main results.** We found the *offline+online* LSTM to perform best for all investigated forecasts. Specifically, when predicting 500 ms ahead it achieved a mean RMSE of 1.20 mm and 1.00 mm, while the best performing LR model achieved a mean RMSE of 1.42 mm and 1.22 mm for the LMU and Gemelli testing set, respectively. **Significance.** This indicates that LSTM networks have potential as respiratory motion predictors and that continuous online re-optimization can enhance their performance.

1. Introduction

Magnetic resonance imaging guided radiotherapy (MR-guided RT) provides radiation-free and high soft tissue contrast imaging, allowing for inter-fractional/intra-fractional motion management and treatment adaptation (Paganelli *et al* 2018b, Kurz *et al* 2020). For tumors affected by respiratory motion such as lung, pancreatic or liver and more generally for malignancies affected by inter-fractional anatomical changes, MR-guided RT offers advantages, such as individualized planning and treatment thanks to its adaptation capabilities (Corradini *et al* 2019, Placidi *et al* 2020).

Both commercially available MR-linacs (Liney *et al* 2018), the MRIdian (ViewRay Inc., Oakwood Village, Ohio, USA) and the Unity (Elekta AB, Stockholm, Sweden) provide the possibility to monitor intra-fractional respiratory motion via 2D+*t* cine MR imaging (Green *et al* 2018, Jackson *et al* 2019, Menten *et al* 2020). For the Unity, recent studies have investigated the usage of multileaf collimator (MLC)-tracking for intra-fractional

motion management, showing its potential to increase the dose delivery accuracy (Menten *et al* 2016) and the technical feasibility of its implementation (Glitzner *et al* 2019, Uijtewaal *et al* 2021). On the other hand, automatic gated beam delivery by use of cine MRI has been clinically used on the MRIdian for years (Green *et al* 2018). To reduce treatment time, gating on the 0.35 T MR-linac is mostly performed in combination with breath-holds. However, not all patients can perform breath-holds or can perform them in a reproducible way (Persson *et al* 2019). MLC-tracking could address this limitation and achieve similar treatment accuracy as gating in a more-time efficient way (Keall *et al* 2021).

To be able to fully exploit the potential of MLC-tracking, the system latency needs to be accounted for in real-time (Poulsen *et al* 2010). The total latency for MLC-tracking is defined as the time lag between the physical target motion and the execution of the MLC motion instructions (Keall *et al* 2021). Recently, Glitzner *et al* have experimentally quantified the total latency for 4 Hz MRI-guidance with MLC-tracking to be about 350 ms for the Elekta Unity (Glitzner *et al* 2019). For the MRIdian, beam-off latencies for gating have been quantified to about 400 ms for the Cobalt-60 version (Green *et al* 2018) and to about 250 ms for the linac version with 4 Hz cine MRI (Kim *et al* 2020). To overcome RT system latencies, several motion prediction algorithms have been proposed in the past (Sharp *et al* 2004, Krauss *et al* 2011, Yun *et al* 2012): in a recent review study by Joehl *et al* a continuously re-optimized (i.e. online) linear regression (LR) model was found to perform best on average compared to other motion predictors such as artificial neural networks or Kalman filters (Joehl *et al* 2020).

In the past few years, several different artificial intelligence (AI) algorithms have found relevant applications in the field of MR-guided RT, e.g. in image segmentation, synthetic CT reconstruction or automatic online planning (Cusumano *et al* 2021). Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997) are a class of AI models which were designed to efficiently capture temporal dependencies in the input data and are therefore ideally suited for motion prediction. In fact, studies have shown the potential of LSTMs for motion prediction in RT based on infrared real-time position management data from a standard linear accelerator (Lin *et al* 2019) and on optical fiducial marker data from a robotic radiosurgery system (Wang *et al* 2018).

In this work, we developed LSTM networks and benchmarked their performance with LR models for the prediction of tumor centroid positions based on 4 Hz cine MRI data acquired at two different institutes with a MRIdian MR-Linac and a MRIdian MR-Cobalt-60. Specifically, motion curves from patients treated at the University Hospital of the LMU Munich were used for training, validation and testing of the models. Additionally, patients treated at the Fondazione Policlinico Universitario Agostino Gemelli in Rome were used as independent testing set. Both the LSTM and the LR were implemented with offline and online training schemes, taking into account feasibility in a 4 Hz intra-fractional motion management clinical scenario. To the best of our knowledge, this is the first study in which LSTMs were applied to MR-guided RT data and in which the usage of continuously re-optimized LSTMs was investigated for motion prediction in RT.

2. Material and methods

2.1. Respiratory motion data

We retrospectively collected respiratory motion data from 2D+t cine MRIs across two institutions. Specifically, cine videos from 88 patients were collected at the Department of Radiation Oncology of the University Hospital of the LMU Munich. As the RT treatment received by every patient is usually split into several fractions and for each fraction a cine MRI sequence is acquired, we obtained 556 videos from the 88 LMU patients. All patients were treated with the MRIdian MR-linac using breath-hold techniques and comprised tumors in the lung (37 cases), pancreas (22), heart (6), liver (20) and mediastinum (3). At the Fondazione Policlinico Universitario Agostino Gemelli in Rome, three patients with in total 15 cine videos were collected. For this cohort, we only selected patients treated in free-breathing using a MRIdian MR-Cobalt-60 machine. Tumor sites comprised lung (2) and pancreas (1).

For all cohorts, the 2D+t cine MRs were acquired at 4 Hz in a sagittal plane with a balanced steady-state free precession sequence (TRUFI; in-plane resolution $3.5 \times 3.5 \text{ mm}^2$; field-of-view $270 \times 270 \text{ mm}^2$ or $350 \times 350 \text{ mm}^2$; slice thickness of 5, 7 or 10 mm). The information on the field-of-view was used to convert the motion amplitudes from video pixels into mm. The cine MRs were exported with target and boundary contours in the OGV video format, as supported by the vendor. This resampled and interpolated video file was then used for analysis. The contours are present in every exported cine MR frame as they are used for the gated beam delivery: prior to treatment, a user defines a target structure (tumor) in a sagittal slice of the volumetric MRI, as well as a boundary structure which defines the gating area where the beam is turned on. During treatment, the target contour is continuously propagated to the current cine MR frame using fast deformable image registration by the vendor's software (Green *et al* 2018, Klueter 2019).

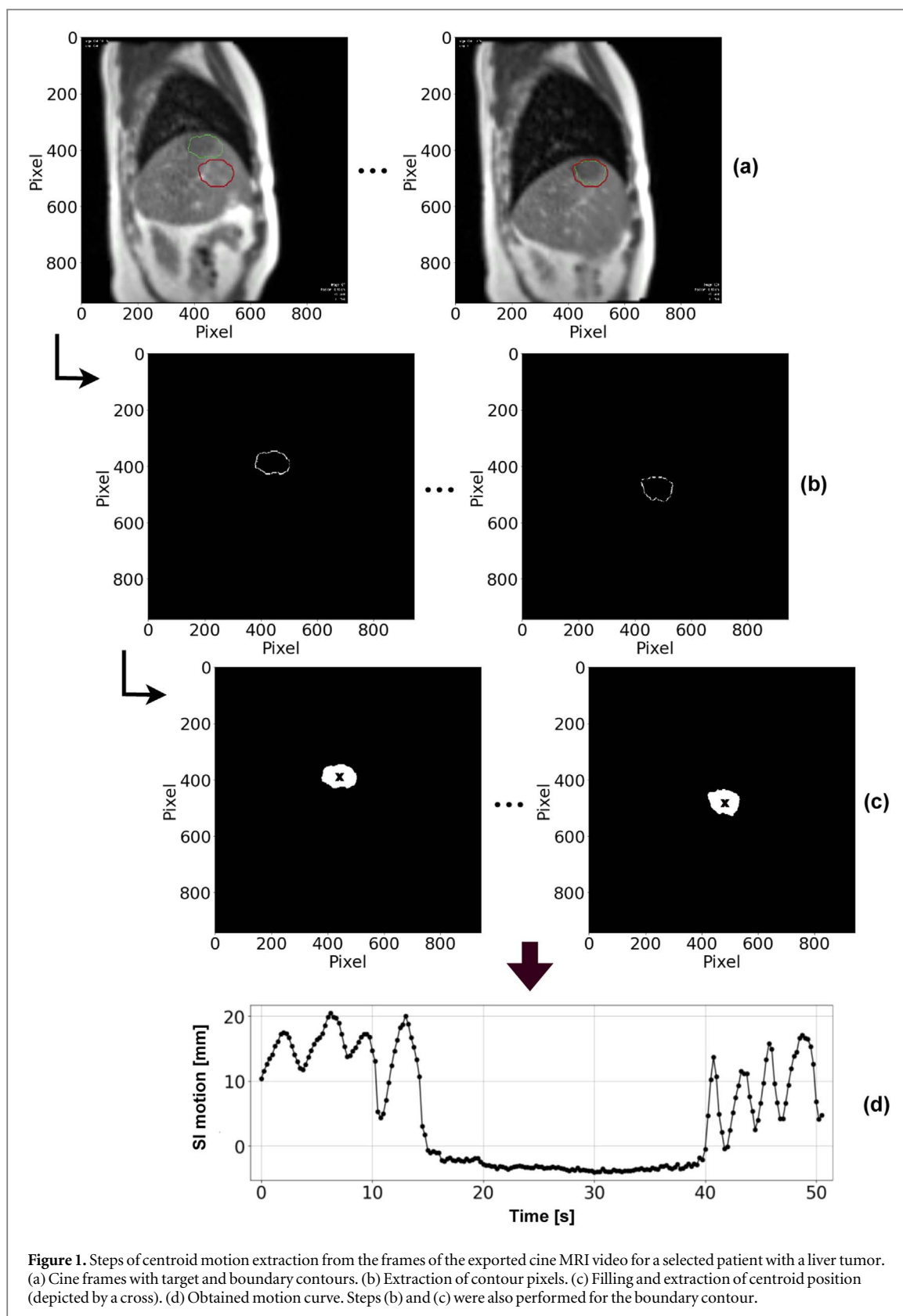


Figure 1. Steps of centroid motion extraction from the frames of the exported cine MRI video for a selected patient with a liver tumor. (a) Cine frames with target and boundary contours. (b) Extraction of contour pixels. (c) Filling and extraction of centroid position (depicted by a cross). (d) Obtained motion curve. Steps (b) and (c) were also performed for the boundary contour.

2.2. Data pre-processing

2.2.1. Centroid position extraction

To obtain motion trajectories from the cine MRIs we used an in-house developed software. Figure 1 summarizes the workflow of the motion extraction from the cine MR frames containing target (green) and boundary (red) contours (a). Briefly, both target and boundary contours were extracted from the videos using thresholds in RGB-space (b). The contours were then filled using the watershed algorithm (Roerdink and Meijster 2000). From the filled contours we subsequently computed the superior–inferior (SI) tumor centroid position relative

to the fixed boundary SI centroid position (c). Once the tumor SI motion curves (d) were obtained for all patients, further pre-processing was done, as detailed in the following sections.

2.2.2. Outlier replacement and filtering

First, we replaced outliers arising from incorrect filling of contours using sliding windows of size three. In detail, we computed the median centroid position within the current sliding window and if the absolute difference between the central data point in the window and the median value was a larger than an optimized threshold, we replaced that point with the median value of the window. For this step, the curves were temporarily normalized such that a single threshold independent from the absolute motion amplitudes could be used. The normalization was then reversed as the motion in mm is needed to exclude cine videos with small motion (see 2.2.4). After that, the curves were smoothed with a moving average filter applied on sliding windows of size three.

2.2.3. Breath-holds and image pauses exclusion

For the LMU cohort only, we then analyzed the motion trajectories to detect the breath-holds. Using sliding windows of size 20, we considered the current window a breath-hold if the median deviation from the median centroid position of the window was smaller than an optimized threshold. This information was then used to exclude all breath-hold data points from the motion trajectories and keep only the free-breathing sub-trajectories in between. Additionally, for this cohort we separated the data according to detected imaging pauses. Cine imaging pauses are inherently part of the MRIdian MR-linac treatment: when the gantry rotates from one irradiation angle to the next, its moving electronics interferes with the MRI causing the image quality to degrade. These degraded cine MRI frames are automatically excluded from the exported videos by the vendor, but their start is indicated by displaying the statement ‘imaging paused’ on the top right of the video. We automatically detected the frames where this statement was displayed and used this information to separate the motion trajectories into two sub-trajectories. This avoids jumps in the curves arising from an imaging pause between data points.

2.2.4. Small motion exclusion and data normalization

For both the LMU and the Gemelli cohorts we excluded all data of cine videos for which the interquartile range (IQR) of SI free-breathing motion was below 3.5 mm (in-plane resolution of MRIdian cine MRIs), as this motion is more substantially affected by imaging noise. This led to the exclusion of 73 LMU and 5 Gemelli videos. Finally, we again normalized all motion curves to the range -1 to $+1$ using the minimum and maximum tumor centroid position of each cine MRI. These min/max values were saved to disk and used during evaluation to undo the normalization of the predicted curves.

After pre-processing, we obtained 16.1 h of motion data without breath-holds (105.8 h if the breath-holds were not excluded) for the LMU cohort and 1.5 h of free-breathing motion for the Gemelli cohort.

2.3. Motion prediction models

2.3.1. Mathematical formulation of prediction problem

Following the terminology used by Remy *et al* (2021), motion prediction is about obtaining future target positions (at time $t + \Delta t$) from the current target motion (at time t). In general, for a given time step i , every prediction task can be simply formulated as

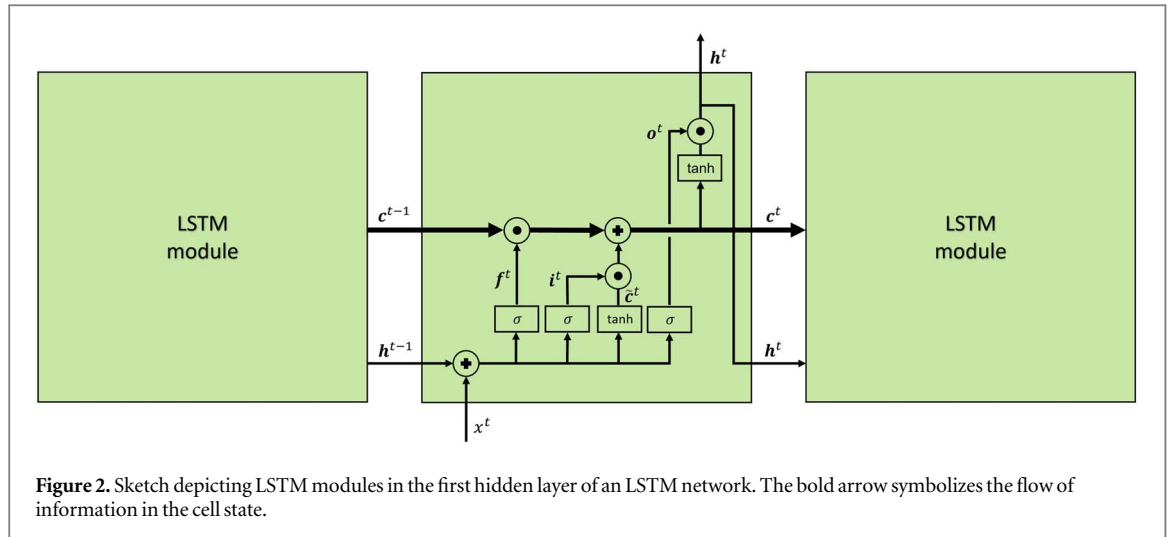
$$\hat{y}_i = f(\mathbf{x}_i), \quad (1)$$

where $f()$ is a motion prediction algorithm, \mathbf{x}_i is the i th vector containing the input data window and \hat{y}_i is the i th vector with the predicted output data window. The corresponding vector \mathbf{y}_i contains the ground truth output data window used to optimize the algorithm. In our case, \mathbf{x} and \mathbf{y} contain input and output SI target centroid positions. The length of \mathbf{x} was treated as a hyper-parameter (see section 2.4.2). On the other hand, the length of \hat{y} (and \mathbf{y}) is automatically related to the forecasted time span. In this study, we investigated forecasts of 250 ms, 500 ms and 750 ms, corresponding to \hat{y} having length of 1, 2 or 3, respectively, for 4 Hz imaging.

2.3.2. Linear ridge regression

Over the last decade, several motion prediction algorithms have been proposed to account for latencies in image guided RT. Joehl *et al* used motion traces from a robotic radiosurgery system to compare 18 different predictors for 160 ms and 480 ms forecasts (Joehl *et al* 2020). On average, they found an (online) LR model to perform best, so we decided to leverage it as baseline model. Mathematically, the regression function is defined as (Krauss *et al* 2011)

$$f(\mathbf{x}) = \beta^T \mathbf{x} + \beta_0, \quad (2)$$



where the vector β contains the parameters of the regression model. The loss function to be minimized to solve the regression is given by

$$L(\beta) = \sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2 + \lambda \|\beta\|^2, \quad (3)$$

where N is the number of input/output training windows and λ is an L2-regularization parameter. If $\lambda \neq 0$, the term ridge regression is usually used.

If we define a matrix \mathbf{X} such that its rows equal the input window vectors \mathbf{x}_i and \mathbf{Y} a matrix with the true output window vectors \mathbf{y}_i , the loss function $L(\beta)$ is analytically solved by the optimal parameters

$$\beta^* = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{Y}, \quad (4)$$

where \mathbf{I} is the identity matrix. Therefore, the LR model has a closed form solution and does not need iterative optimization.

2.3.3. LSTM

Recurrent neural networks (RNN) are a class of machine learning algorithms ideally suited for sequential input data (e.g. time series data). Compared to artificial neural networks or convolutional neural networks, recurrent neural networks have an extra set of weights which connects hidden layers from one time point to the next. However, the original RNN module comprising a simple fully connected layer was found to be limited due to unstable gradient issues when back-propagating over longer input sequences (Shen *et al* 2020). Therefore, more advanced RNN architectures have been proposed and the most widely adopted is the LSTM model (Hochreiter and Schmidhuber 1997), which was specifically designed to more easily learn longer sequences of data.

The repeating module of LSTMs is shown in figure 2. LSTMs introduce the memory cell state \mathbf{c}^t , which allows a stable back-propagation of errors and straightforward flow of information. To remove or add information to the cell state, structures called gates are used. Intuitively, the forget gate \mathbf{f}^t is used to keep/discard past information in \mathbf{c}^{t-1} while the input gate \mathbf{i}^t allows to filter information in the new memory cell state $\tilde{\mathbf{c}}^t$. The previous gates and memory are then combined to build the final memory cell state \mathbf{c}^t . The final memory cell is filtered with the output gate \mathbf{o}^t to keep/discard some information and build the hidden state \mathbf{h}^t . Mathematically, at a specific time step t , the LSTM module is described as follows:

$$\text{Forget gate: } \mathbf{f}^t = \sigma(\mathbf{W}_f \mathbf{x}^t + \mathbf{U}_f \mathbf{h}^{t-1} + \mathbf{b}_f) \quad (5)$$

$$\text{Input gate: } \mathbf{i}^t = \sigma(\mathbf{W}_i \mathbf{x}^t + \mathbf{U}_i \mathbf{h}^{t-1} + \mathbf{b}_i) \quad (6)$$

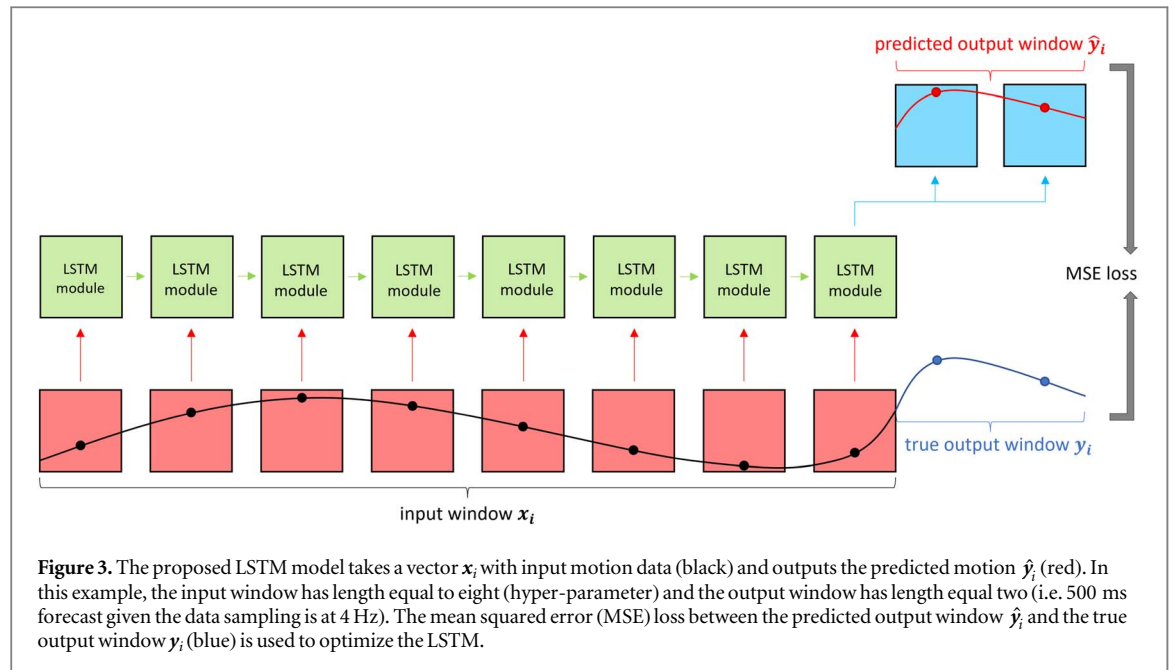
$$\text{New memory cell state: } \tilde{\mathbf{c}}^t = \tanh(\mathbf{W}_c \mathbf{x}^t + \mathbf{U}_c \mathbf{h}^{t-1} + \mathbf{b}_c) \quad (7)$$

$$\text{Final memory cell state: } \mathbf{c}^t = \mathbf{f}^t \odot \mathbf{c}^{t-1} + \mathbf{i}^t \odot \tilde{\mathbf{c}}^t \quad (8)$$

$$\text{Output gate: } \mathbf{o}^t = \sigma(\mathbf{W}_o \mathbf{x}^t + \mathbf{U}_o \mathbf{h}^{t-1} + \mathbf{b}_o) \quad (9)$$

$$\text{Hidden state: } \mathbf{h}^t = \mathbf{o}^t \odot \tanh(\mathbf{c}^t), \quad (10)$$

where \mathbf{b} , \mathbf{W} and \mathbf{U} denote the biases, input window weights and recurrent weights which are learned during the optimization process. The symbol \odot represents element-wise multiplication between matrices/vectors. The sigmoid function $\sigma(x)$ was used for the gates and the hyperbolic tangent function was used to generate the states. For each time step, the hidden state of one LSTM layer is used as input for the next LSTM layer. For the last



hidden layer of the LSTM, the hidden state of the last time point t_f is input to a fully connected layer to build the predicted output window

$$\text{Predicted output: } \hat{y}_i = \mathbf{W}_{\text{FC}} \mathbf{h}^{t_f} + \mathbf{b}_{\text{FC}}, \quad (11)$$

where \mathbf{W}_{FC} and \mathbf{b}_{FC} denote the weight matrix and bias vector for the fully connected layer.

In this study, a stateless LSTM was implemented, which means that during optimization the hidden state and the cell state were cleared after every batch of data. The LSTM architecture was inspired by the one used by Lin *et al* (2019). Specifically, we performed our hyper-parameter optimization based on the range of values used in their hyper-parameter search. More details can be found in section 2.4. Figure 3 schematically shows the working principle of the proposed LSTM. At every time point, a single SI tumor centroid position is given as input for as many points as the length of the input data window. The LSTM modules in the hidden layer (green boxes) process the time-dependent information as shown in figure 2 until the last LSTM module is reached. The hidden vector which is output by the last LSTM module is mapped via a fully connected layer to the predicted output window following equation (11). Note that in figure 3 only one hidden layer is shown whereas this number was treated as a hyper-parameter in our optimizations (see section 2.4.2).

2.4. Model optimization

2.4.1. Data subdivision

To optimize and evaluate the models we split the LMU data into training, validation and testing sets. Specifically, we assigned the motion trajectories belonging to 60% of the patients to the training set (52 patients), 20% to the validation set (18 patients) and the remaining 20% to the testing set (18 patients) and did this procedure only once at the beginning. This splitting roughly also led to 60% of the motion trajectories being in training (9.1 h), 20% in validation (4.0 h) and 20% in testing (3.0 h). As the Gemelli cohort was smaller (1.5 h) but at the same time in free-breathing, we decided to use this dataset as an independent additional testing set. Finally, we also applied the best models trained/validated on the LMU data without breath-holds to the LMU testing set without excluding the breath-holds during pre-processing.

2.4.2. Hyper-parameter search

To find the optimal set of hyper-parameters for both the LSTM and the LR we repeatedly performed training and validation while varying the parameters for all three analyzed forecasts and for all four training strategies (see section 2.4.3) separately. For the LSTM, the following hyper-parameters were varied, based on the hyper-parameter search performed by Lin *et al*:

- *Number of layers*: the number of hidden layers of the LSTM was chosen among the following values {1, 3, 5, 10}.
- *Dropout*: the dropout rate on the outputs of each hidden layer (but the last one) was sampled from the set {0, 0.1, 0.2}.

- *Learning rate*: the learning rate of the optimizer was sampled from the set $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}\}$.
- *Batch size*: the batch size, i.e. the number of input windows fed to the network simultaneously, can be varied only for the *offline* trained LSTM (see section 2.4.3) and was set to either 64 or 128.
- *L2-regularization*: the L2-regularization parameter λ (also called weight decay) was sampled from the set $\{0, 1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}\}$.

All optimizations for the LSTMs were carried out using the Adam optimizer (Kingma and Ba 2015) with a normalized mean squared error (MSE) loss function and learning rates from the set shown above. We set the number of features in the hidden layer vector \mathbf{h}^t to 15 like Lin *et al*. No batch normalization was used. For the LR, the following hyper-parameter was varied in logarithmic steps over a large range of values:

- *L2-regularization*: the L2-regularization parameter λ was sampled from the set $\{1 \times 10^{-5}, 1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}, 1, 10\}$.

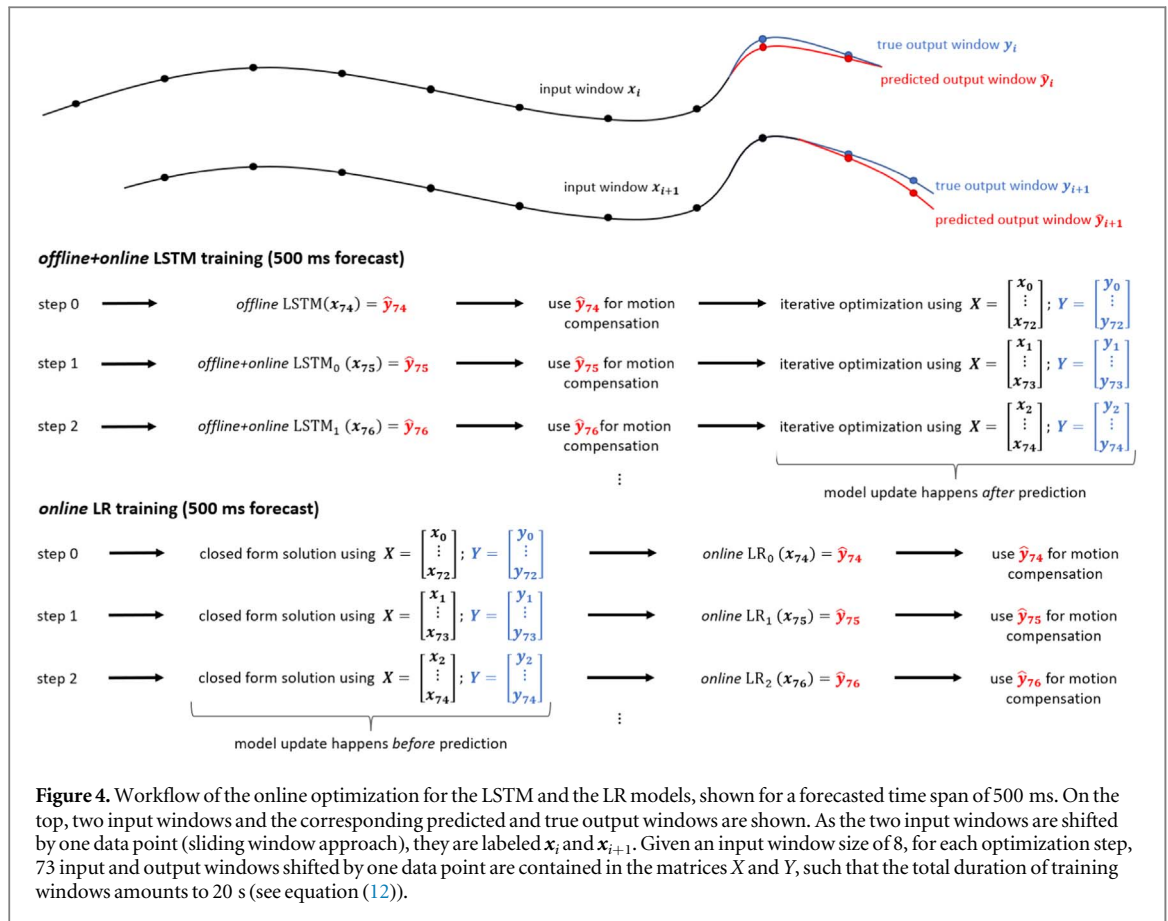
Both for the LSTM and for the LR we varied the length of the input data window \mathbf{x} between 8, 16, 24 and 32 data points, corresponding to 2, 4, 6 and 8 seconds of past motion.

2.4.3. Training strategies

Retraining models on recent motion data has been shown to improve the predictive performance (Krauss *et al* 2011, Sun *et al* 2020). Thus, two different training strategies were investigated for the LSTM and for the LR model.

- *Offline LSTM*: the *offline* LSTM optimization was carried out following the typical machine learning training/validation/testing subdivision. The model was iteratively optimized on the training set for 600 epochs while monitoring training and validation losses. If the validation loss improved, the model weights were saved to disk. If the validation loss did not improve for 100 epochs, the optimization was stopped, a technique known as early stopping. For final inference, we loaded the weights and hyper-parameters of the best performing model on the validation set and applied it unchanged to the testing set.
- *Offline+online LSTM*: to allow adaptation to recent motion patterns, we continuously retrained the LSTM on current data. Specifically, we first loaded the weights of a previously optimized *offline* LSTM. The LSTM was then re-optimized on the last 20 s of validation data using a sliding set of validation input/output windows updated with a first-in-first-out approach, as shown in figure 4. The online optimization of the LSTM was done for 10 epochs, taking about 150 ms. This would allow an implementation in a 4 Hz image acquisition clinical scenario. To prevent the iterative optimization to introduce an additional latency, within the 250 ms between one cine MRI frame and the next, we performed the prediction *before* optimizing the LSTM. To calculate the validation loss we used the ground truth data point lying 250 ms, 500 ms or 750 ms (depending on the forecast) in the future with respect to the last centroid position in the currently used 20 s of optimization data. For final inference, we loaded the *offline* LSTM, set the hyper-parameters leading to the best result on the validation set and continuously retrained and evaluated the model on the testing set.
- *Offline LR*: the *offline* LR training is analogous to the *offline* LSTM training but for the fact the the LR is solved analytically while the LSTM is iteratively optimized. Specifically, the LR was solved on the training set and then applied unchanged to validation set to perform the hyper-parameter search. For final inference, as for the *offline* LSTM, we loaded the weights and set the hyper-parameters of the best performing model on the validation set and applied it unchanged to the testing set.
- *Online LR*: on the other hand, the *online* LR is different from *offline+online* LSTM. As no iterative fine-tuning is needed for the LR, no weights from a pre-trained *offline* LR were loaded. The *online* LR was continuously solved ‘from scratch’ based on the last 20 s of validation data using a sliding set of validation input/output windows updated with a first-in-first-out approach (figure 4). As solving the LR is simply a matrix multiplication (see equation (2)), it takes less than 1 ms. As this additional latency is not significant, for the *online* LR we performed the prediction *after* the optimization, as illustrated in figure 4. This is advantageous as the model’s prediction can take into account the most recently acquired data point.

As mentioned in section 2.3.1, our data was subdivided in input windows \mathbf{x} where the number of entries $\text{len}(\mathbf{x})$ is a hyper-parameter. To obtain a set of windows with a total duration of 20 s to be used for online training, we need several input windows. Given that every input window is shifted by one and that the cine imaging is



performed at a frame rate of 4 Hz, the number of input windows needed is given by

$$\text{Nr. input windows} = 20 \cdot 4 - \text{len}(x) - 1. \quad (12)$$

Therefore, 73 input windows are needed if we choose $\text{len}(x) = 8$ as in figure 4. This number corresponds to the batch size for the *offline+online* LSTM, which is thus not freely selectable if we fix the duration of the online optimization data to 20 s. The number of input windows between the last window used for optimization and the window used for prediction is given by the current forecasted time span. For example for the 500 ms forecast scenario shown in figure 4, this difference is equal to two (e.g. step 0: x_{72} is the last window in the optimization matrix and x_{74} is the window used for prediction). For the 250 ms forecast this difference is one and for the 750 ms forecast this difference is equal to three.

2.5. Loss and evaluation metrics

The loss function used to optimize the LSTM was the MSE, which is defined as

$$\text{MSE} = \frac{1}{B} \sum_{i=1}^B (y_i - \hat{y}_i)^2, \quad (13)$$

where B is the batch size, y_i is the vector with the true output window and \hat{y}_i is the vector with the predicted output window of centroid positions. Note that the MSE was computed using normalized output windows.

The root mean squared error (RMSE) and maximum error (ME) error were used to evaluate the LSTM and LR predictive performance on the validation and testing sets. Prior to the computation of the evaluation metrics, the normalization of the ground truth and predicted curves was reversed, such that the metrics are in mm. The RMSE and ME were calculated on a treatment fraction basis (one RMSE/ME per cine MRI video) and are defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (14)$$

$$\text{ME} = \max\{|y_i - \hat{y}_i|, i = 1, 2, \dots, N\}, \quad (15)$$

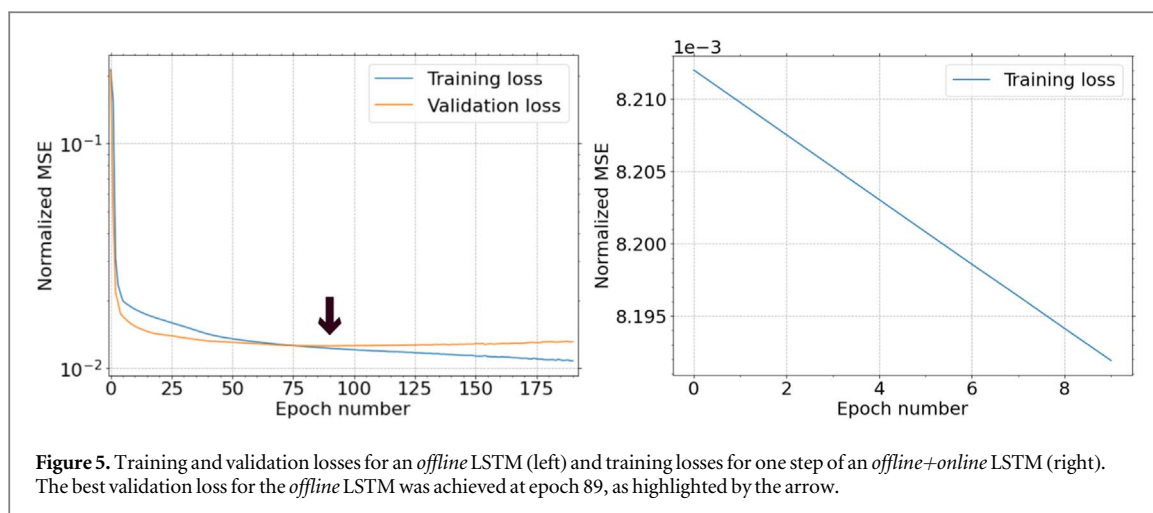


Figure 5. Training and validation losses for an *offline* LSTM (left) and training losses for one step of an *offline+online* LSTM (right). The best validation loss for the *offline* LSTM was achieved at epoch 89, as highlighted by the arrow.

where N is the number of data points in the motion curves belonging to a single cine MRI, y_i is the true future centroid position and \hat{y}_i is the predicted centroid position. The analysis was done using only the last element of each output window, which is the hardest to predict (Sharp *et al* 2004).

Finally, we averaged over the RMSEs/MEs of different fractions to build the mean RMSE/ME with corresponding standard deviation.

2.6. Statistical tests

To analyze if there is a statistically significant difference between the RMSE values obtained with the different models on the different testing sets, non-parametric Friedman tests were performed (Friedman 1937). A p -value < 0.05 was considered significant. If the Friedman test revealed a significant difference, we consecutively performed a post-hoc Nemenyi test (Nemenyi 1963) to infer which model obtained significantly better RMSEs in a pair-wise fashion.

2.7. Implementation details

All code used for this study was written in Python 3.8.5 and is freely available: https://github.com/LMUK-RADONC-PHYS-RES/lstm_centroid_prediction. To build and optimize the LSTMs, the PyTorch library (Paszke *et al* 2017) version 1.8.0 was used. Training for both the *offline* and the *offline+online* LSTM was carried out on an NVIDIA Quadro RTX 8000 GPU with 48 GB of memory. The LR was built and solved using the scikit-learn library (Pedregosa *et al* 2011) version 0.24.1. The LR was trained on an Intel Xeon Gold 6254 (Cascade Lake-EP) 18-Core CPU.

3. Results

In terms of prediction speed, a forward pass with an LSTM takes about 5 ms while for the LR models less than 1 ms.

3.1. Validation

Figure 5 shows the normalized MSE losses for an optimization of an *offline* LSTM and an *offline+online* LSTM. For the shown *offline* LSTM, the best validation loss was obtained at epoch 89 which led to early stopping of the optimization at epoch 189. On the other hand, no validation loss was monitored for the *offline+online* LSTM. As shown in figure 4, within one training step we first performed the prediction and then iteratively re-optimized the LSTM for 10 epochs (see section 2.4.3), as this is the maximum number of epochs which would still allow a re-optimization in a 4 Hz clinical scenario.

Table 1 shows the best RMSEs obtained with the four different models on the validation set. The corresponding set of best hyper-parameters for each model is shown in the appendix (tables A1, A2 and A3). For all three forecasted time spans, the *offline+online* LSTM achieved the best performance, reaching basically the same as the *offline* LSTM for the 250 ms forecast, and slightly better performance for the 500 ms and 750 ms forecasts. The best performing LR was the *offline* one, however its performance was worse than both LSTM training schemes.

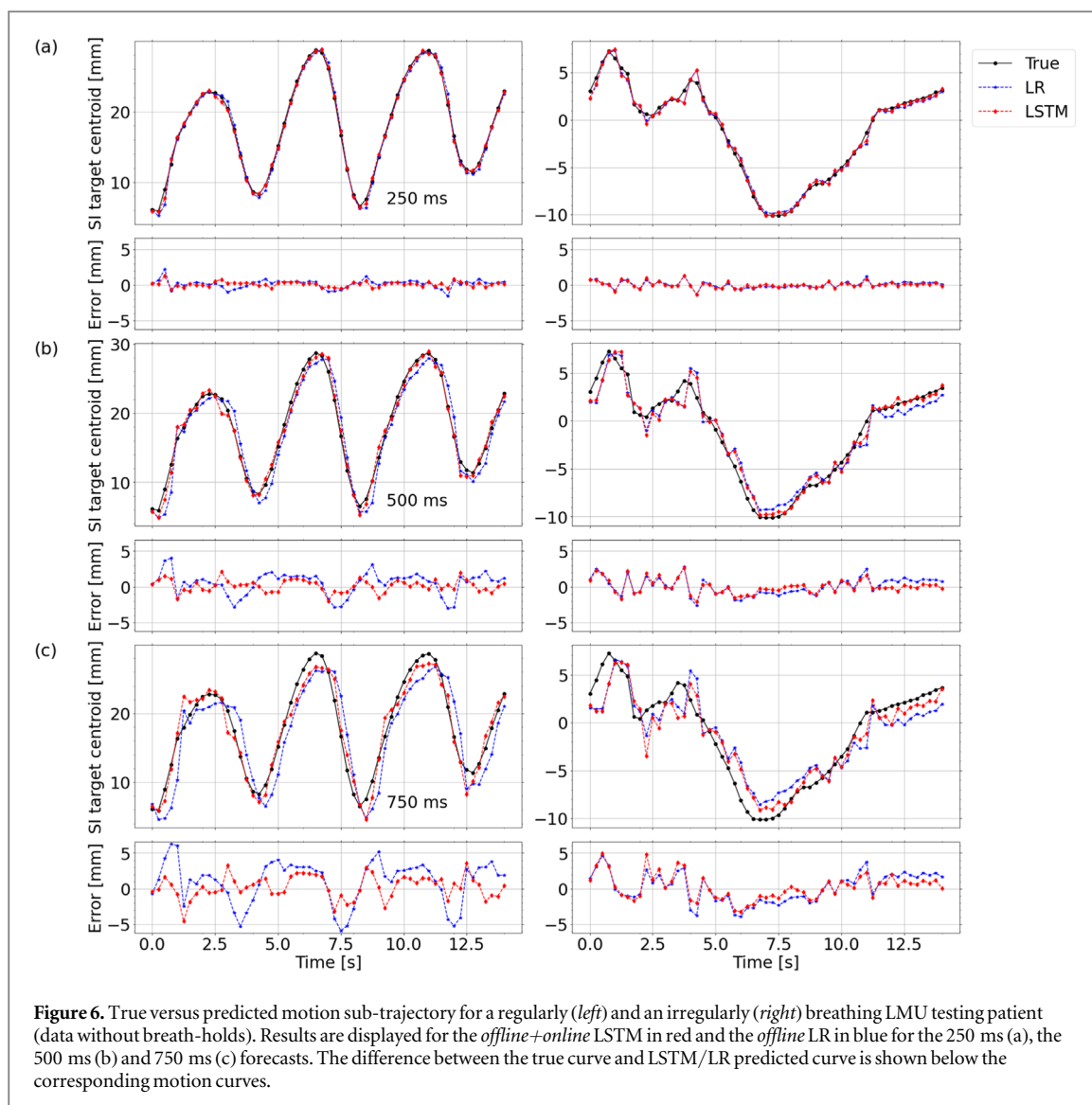


Table 1. Mean and standard deviation of RMSEs for the validation set. The RMSE of the best performing model is shown in bold for each forecasted time span.

| Model | 250 ms forecast | 500 ms forecast RMSE [mm] | 750 ms forecast |
|----------------------------|--------------------|------------------------------|--------------------|
| <i>Offline</i> LSTM | 0.55 ± 0.44 | 1.40 ± 1.00 | 2.58 ± 1.71 |
| <i>Offline+online</i> LSTM | 0.54 ± 0.43 | 1.36 ± 0.94 | 2.54 ± 1.63 |
| <i>Offline</i> LR | 0.63 ± 0.49 | 1.68 ± 1.11 | 3.09 ± 1.91 |
| <i>Online</i> LR | 0.74 ± 0.53 | 1.76 ± 1.13 | 3.15 ± 1.87 |

3.2. Testing

Figure 6 shows for two selected patients of the LMU testing set (data without breath-holds) ground truth versus predicted respiratory motion trajectories for the best LSTM model (*offline+online*) and the best LR model (*offline* LR) for all the forecasted time spans. Qualitatively, no noticeable difference is seen when comparing the best LR and the best LSTM for the 250 ms forecast. On the other hand, both for the 500 ms forecast and for the 750 ms one can see how the LSTM outperforms the LR especially when it comes to predicting steep inhalations/exhalations. Similar observations can be made when looking at figure 7 displaying true vs predicted curves for Gemelli testing patients. Although we noticed that the LSTM overshoots more often than the LR, the former is able to more quickly adapt to changes in the motion trajectories (from steeper/shallower inhalations/exhalations to irregularities) which leads to an overall smaller error, as can be seen in the error plots in figures 6 and 7. Table 2 shows the RMSEs obtained with the four best validation models on the LMU testing set (data without breath-holds). The *offline+online* LSTM was confirmed the best model for all three forecasts. These

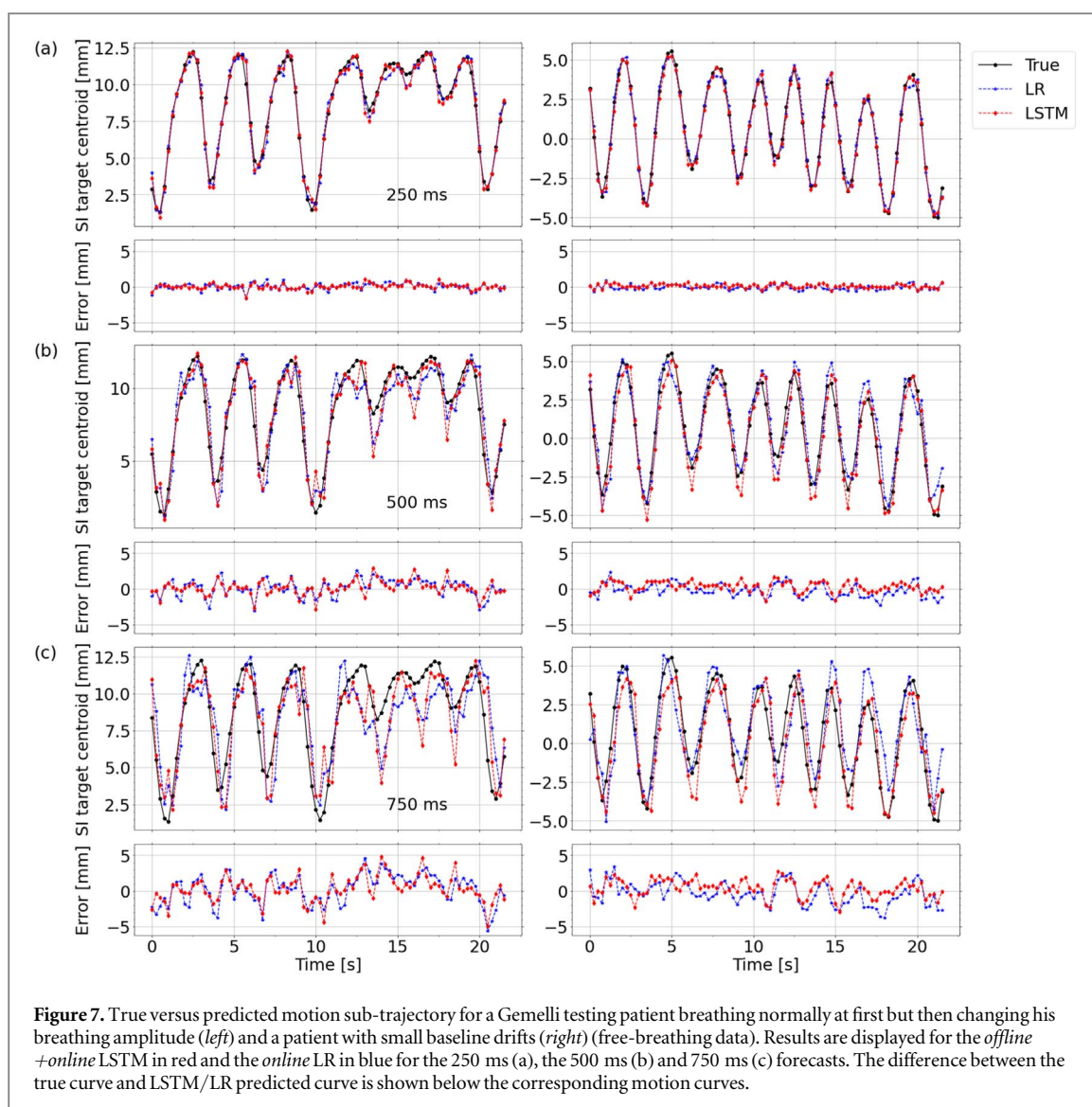


Figure 7. True versus predicted motion sub-trajectory for a Gemelli testing patient breathing normally at first but then changing his breathing amplitude (left) and a patient with small baseline drifts (right) (free-breathing data). Results are displayed for the *offline +online* LSTM in red and the *online* LR in blue for the 250 ms (a), the 500 ms (b) and 750 ms (c) forecasts. The difference between the true curve and LSTM/LR predicted curve is shown below the corresponding motion curves.

Table 2. Mean and standard deviation of RMSEs for the LMU testing set without breath-holds. The RMSE of the best performing model is shown in bold for each forecasted time span.

| Model | 250 ms forecast | 500 ms forecast | 750 ms forecast |
|-----------------------------|--------------------|--------------------|--------------------|
| | RMSE [mm] | | |
| <i>Offline</i> LSTM | 0.49 ± 0.29 | 1.24 ± 0.70 | 2.34 ± 1.25 |
| <i>Offline +online</i> LSTM | 0.48 ± 0.28 | 1.20 ± 0.65 | 2.20 ± 1.12 |
| <i>Offline</i> LR | 0.54 ± 0.30 | 1.42 ± 0.78 | 2.61 ± 1.38 |
| <i>Online</i> LR | 0.64 ± 0.38 | 1.54 ± 0.79 | 2.73 ± 1.41 |

results were confirmed also when looking at MEs, as shown in table A4 in the appendix. In general, all models performed slightly better than during validation both in terms of mean and standard deviation of the RMSE.

When applying the Friedman test, we found a significant difference among the models for all forecasts and testing sets. For the LMU testing set without breath-holds, the post-hoc Nemenyi test yielded the p -values shown in table 3. The best model in terms of RMSEs, i.e. the *offline +online* LSTM was found to perform significantly better than both LR models while there was no significant difference between the *offline* LSTM and the *offline +online* LSTM for all investigated forecasts.

As shown in table 4, the results obtained on the LMU validation and testing set (data without-breath-holds) were confirmed with the Gemelli testing set (free-breathing data). The *offline +online* LSTM was found to perform best for all three forecasted time spans followed by the *offline* LSTM. This time, the *online* LR performed better than the *offline* LR and reached the same RMSE as the *offline* LSTM for the 750 ms forecast. Table 5 shows

Table 3. *P*-values obtained from the post-hoc Nemenyi test for the LMU testing set without breath-holds for all possible pairwise model comparisons. Significant *p*-values ($<5e-2$) are denoted with an asterisk.

| Comparison Model 1 | Model 2 | 250 ms forecast | 500 ms forecast <i>p</i> -value | 750 ms forecast |
|----------------------------|----------------------------|-----------------|------------------------------------|-----------------|
| <i>Offline</i> LSTM | <i>Offline</i> LR | 3e-1 | 3e-2* | 2e-1 |
| <i>Offline</i> LSTM | <i>Online</i> LR | 6e-1 | 1e-1 | 3e-1 |
| <i>Offline+online</i> LSTM | <i>Offline</i> LR | 1e-3* | 1e-3* | 1e-3* |
| <i>Offline+online</i> LSTM | <i>Online</i> LR | 2e-3* | 1e-3* | 1e-3* |
| <i>Offline</i> LSTM | <i>Offline+online</i> LSTM | 9e-2 | 6e-2 | 6e-2 |
| <i>Offline</i> LR | <i>Online</i> LR | 9e-1 | 9e-1 | 9e-1 |

Table 4. Mean and standard deviation of RMSEs for the Gemelli free-breathing testing set. The RMSE of the best performing model is shown in bold for each forecasted time span.

| Model | 250 ms forecast | 500 ms forecast RMSE [mm] | 750 ms forecast |
|----------------------------|--------------------|------------------------------|--------------------|
| <i>Offline</i> LSTM | 0.47 ± 0.12 | 1.14 ± 0.29 | 2.02 ± 0.49 |
| <i>Offline+online</i> LSTM | 0.42 ± 0.13 | 1.00 ± 0.30 | 1.77 ± 0.54 |
| <i>Offline</i> LR | 0.57 ± 0.14 | 1.52 ± 0.34 | 2.76 ± 0.71 |
| <i>Online</i> LR | 0.53 ± 0.17 | 1.22 ± 0.30 | 2.02 ± 0.49 |

Table 5. *P*-values obtained from the post-hoc Nemenyi test for the Gemelli free-breathing testing set for all possible pairwise model comparisons. Significant *p*-values ($<5e-2$) are denoted with an asterisk.

| Comparison Model 1 | Model 2 | 250 ms forecast | 500 ms forecast <i>p</i> -value | 750 ms forecast |
|----------------------------|----------------------------|-----------------|------------------------------------|-----------------|
| <i>Offline</i> LSTM | <i>Offline</i> LR | 3e-2* | 3e-2* | 1e-2 |
| <i>Offline</i> LSTM | <i>Online</i> LR | 2e-1 | 8e-1 | 9e-1 |
| <i>Offline+online</i> LSTM | <i>Offline</i> LR | 1e-3* | 1e-3* | 1e-3* |
| <i>Offline+online</i> LSTM | <i>Online</i> LR | 3e-3* | 2e-2* | 8e-2 |
| <i>Offline</i> LSTM | <i>Offline+online</i> LSTM | 4e-1 | 2e-1 | 8e-2 |
| <i>Offline</i> LR | <i>Online</i> LR | 4e-1 | 6e-2 | 1e-2* |

Table 6. Mean and standard deviation of RMSEs for the LMU testing set with breath-holds. The RMSE of the best performing model is shown in bold for each forecasted time span.

| Model | 250 ms forecast | 500 ms forecast RMSE [mm] | 750 ms forecast |
|----------------------------|--------------------|------------------------------|--------------------|
| <i>Offline</i> LSTM | 0.34 ± 0.17 | 0.83 ± 0.45 | 1.59 ± 0.95 |
| <i>Offline+online</i> LSTM | 0.30 ± 0.17 | 0.74 ± 0.39 | 1.34 ± 0.74 |
| <i>Offline</i> LR | 0.36 ± 0.19 | 0.96 ± 0.51 | 1.83 ± 1.03 |
| <i>Online</i> LR | 0.63 ± 0.65 | 1.39 ± 0.93 | 2.81 ± 2.12 |

the *p*-values obtained with the Nemenyi test on the Gemelli testing results. This time, the *offline+online* LSTM was found to be significantly better than the *offline* LR for all forecasts and significantly better than the *online* LR for the 250 ms and the 500 ms forecast.

Finally, we also applied the best models obtained on the LMU data without breath-holds to the LMU testing set with breath-holds. As shown in table 6, the *offline+online* LSTM again outperformed all other models for all three forecasts followed by the *offline* LSTM and the *offline* LR. While these three models substantially improved their performance compared to the LMU testing set without breath-holds (table 2), the *online* LR even worsened its performance for the 500 ms and 750 ms forecasts. When applying the Nemenyi test, we found significant differences between all pairwise model combinations excluding the *offline* LSTM versus the *offline* LR, as shown in table 7.

Table 7. *P*-values obtained from the post-hoc Nemenyi–Friedman test for the LMU testing set with breath-holds for all possible pairwise model comparisons. Significant *p*-values ($< 5e-2$) are denoted with an asterisk.

| Comparison Model | Model 2 | 250 ms forecast | 500 ms forecast <i>p</i> -value | 750 ms forecast |
|---------------------|---------------------|-----------------|------------------------------------|-----------------|
| Offline LSTM | Offline LR | 3e-1 | 2e-3* | 9e-3* |
| Offline LSTM | Online LR | 1e-3* | 1e-3* | 1e-3* |
| Offline+online LSTM | Offline LR | 1e-3* | 1e-3* | 1e-3* |
| Offline+online LSTM | Online LR | 1e-3* | 1e-3* | 1e-3* |
| Offline LSTM | Offline+Online LSTM | 1e-3* | 1e-3* | 1e-3* |
| Offline LR | Online LR | 1e-3* | 1e-3* | 1e-3* |

Animated figures with sliding input, true output and predicted output windows for a testing patient from the LMU set without breath-holds, the Gemelli set and the LMU set with breath-holds respectively are shown in the online material (stacks.iop.org/PMB/67/095006/mmedia).

4. Discussion

LSTM networks have been successfully applied for time series prediction in many fields, making them one of the most popular versions of RNNs (Shen *et al* 2020). In this study, we applied LSTMs to forecast tumor centroid positions based on respiratory motion trajectories obtained from 0.35 T MR-linacs. The fact that the proposed *offline+online* LSTM outperformed all the other models for all testing cohorts and for all forecasted time spans confirms our hypothesis that LSTMs are well suited for motion prediction in MR-guided RT. The *offline+online* LSTM was found to perform significantly better than the best performing LR in 8/9 testing scenarios. The only scenario where the better RMSE of the *offline+online* LSTM was not significant compared to the best LR was for the 750 ms forecast with the Gemelli testing data. However, the Gemelli testing set presents less data compared to the other two testing sets, as there are less videos over which the RMSE is calculated.

As expected from literature (Murphy and Pokhrel 2009, Sun *et al* 2020), the *offline+online* LSTM achieved better performance than the *offline* LSTM for all testing cohorts when looking at RMSE. Additionally, for the LMU testing set with breath-holds this difference was statistically significant. As this testing cohort differs more substantially from the training and validation sets than the other two, we expected this improvement. In general, we conclude that iterative fine-tuning using the latest respiratory patterns is beneficial also for LSTMs. The *offline+online* LSTM was implemented such that online optimization took about 150 ms and could therefore be used in a 4 Hz cine MRI guided RT treatment.

All models achieved better mean RMSE on the LMU testing set with breath-holds included. This is expected since this data contains long time intervals of flat motion trajectories, which are easy to predict.

For the LMU testing set without breath-holds, the *offline* LR regression was found to perform better than the *online* LR (see table 2), a finding in disagreement with literature (Krauss *et al* 2011, Uijtewaal *et al* 2021). However, the difference was not significant, as shown in table 3. When comparing the mean RMSE obtained for the 500 ms forecast with our *offline* LR to the mean RMSE obtained with the *online* LR by Uijtewaal *et al* for the 500 ms forecast, we can see that both models achieved a value of about 1.5 mm. Furthermore, the *online* LR was found to perform better than the *offline* on the Gemelli testing set, which likely differs from the LMU training set. Additionally, the free-breathing Gemelli data might be easier to predict than the LMU data without breath-holds as the latter consists of sub-trajectories of free-breathing motion in-between breath-holds and can thus contain irregular breathing or steep inhalations and exhalations. This could also explain why the *online* LR performed better on the Gemelli testing set.

To compare the performance obtained in this study with the one obtained by the LSTM implemented by Lin *et al*, we report here normalized RMSEs obtained with our *offline+online* LSTM for the 500 ms forecast. The normalization was done using the min-max amplitudes saved to disk during pre-processing. We found a mean normalized RMSE of 0.086 for the LMU testing set without breath-holds and 0.107 for the Gemelli testing set. These results are in agreement with the mean testing RMSE of 0.139 found by Lin *et al* (2019). Furthermore, we can approximately compare the RMSE obtained with our *offline+online* LSTM for the 500ms forecast with the RMSE obtained by Wang *et al* using a Bi-LSTM for a 400 ms forecast (Wang *et al* 2018). Since they found a mean validation normalized RMSE of 0.081 (no testing set was used, unlike in this study), we conclude that our *offline+online* LSTM is comparable.

In general, we noticed large standard deviations of the RMSEs. This suggests that substantial performance differences might be observed among different patients. The standard deviations for the Gemelli testing set and the LMU testing set with breath-holds were smaller than for the LMU testing set without breath-holds. As the Gemelli data consists of regular free-breathing and the LMU data with breath-holds largely consists of flat

motion regions, we hypothesize that this decreased variability in the data leads to smaller standard deviations. The fact that the mean and standard deviation of the validation RMSE were larger than the LMU testing results (compare tables 1 and 2), might be explained by the fact that by chance, when splitting the LMU cohort in training, validation and testing, more irregular motion curves were assigned to the validation compared to the testing set.

As observed in several studies (Sharp *et al* 2004, Seregni *et al* 2016, Wang *et al* 2018, Uijtewaal *et al* 2021), the predictive performance decreased with increasing forecasted time span. However, sub-resolution accuracy (<3.5 mm) was still reached for all three forecasts. The good RMSE of about 1 mm of the *offline+online* LSTM for the 500 ms forecast shows that this model could be used to successfully account for the system latencies found by Glitzner *et al* (2019) when performing MLC tracking on an Elekta Unity MR-linac.

The current study has a few limitations. The first is that all models were optimized and applied on motion curves which were normalized based on the global minimum and maximum SI centroid position of each cine video, following (Lin *et al* 2019, Yu *et al* 2020). In clinical practice of course, the global minimum and maximum for the entire fraction cannot be known before the treatment ends. However, with the 0.35 T MR-linac, right before the treatment starts a preview cine MRI is acquired (Klueter 2019) for automatic selection of a tracking key frame, to inspect if the gating window needs to be adjusted and similar aspects. This cine MRI could also be used to get the min-max amplitudes to be used for the normalization of the motion curves acquired for the treatment. A small window size (equal to three) was taken for both the outlier replacement and the moving average filter to make an implementation in a real-time clinical scenario possible. The second limitation consists in the fact that our models only predict the future centroid position in SI direction. While this could already be used for centroid MLC tracking in parallel direction, where the MLC shape is shifted to the predicted SI position, latencies for deviations in anterior-posterior direction would not be accounted for. To achieve this, a second model predicting the other direction could be run in parallel. Alternatively, the anterior-posterior motion could be included as input, a possible extension to the models presented in this work. However, only predicting centroid positions would not allow for more advanced forms of dynamic MLC tracking (Ge *et al* 2014), where the MLC shape is adapted the predicted tumor location and shape, possibly taking into account in-plane rotations and deformations (Keall *et al* 2021). In a future study, we plan to extend the proposed LSTM to directly predict future 2D cine MR frames, thus allowing for dynamic MLC tracking. Finally, our model cannot predict out-of-plane motion as the cine MRIs are acquired in a single sagittal plane. However, several methods have been proposed to obtain time-resolved volumetric MRI (Fayad *et al* 2012, Stemkens *et al* 2016, Paganelli *et al* 2018a, Rabe *et al* 2021), which might be combined with our motion prediction model in future studies.

5. Conclusions

In this study, we developed LSTMs for SI tumor centroid position prediction based on cine MRIs acquired with 0.35 T MRIdian machines from two different institutions and showed that they outperformed state-of-the-art LR algorithms for all investigated forecasts (250 ms, 500 ms and 750 ms). The proposed models generalized their predictive performance to different testing sets with different breathing patterns, ranging from free-breathing to treatments with prolonged breath-holds. The continuously re-optimized *offline+online* LSTM network achieved superior performance in all tasks compared to offline optimized models. In conclusion, LSTMs were shown to have great potential as respiratory motion predictors to account for the system latencies present in MR-guided RT with MLC tracking.

Acknowledgments

Henning Schmitz is thanked for support related to the manuscript preparation, Andreas Haslauer for initial work on the cine videos and Seyed-Ahmad Ahmadi for fruitful discussions. This work was supported by the German Research Foundation (DFG) within the Research Training Group GRK 2274.

Conflict of interest

The Department of Radiation Oncology of the University Hospital of the LMU Munich has research agreements with Brainlab, Elekta and ViewRay.

Appendix

Table A1. Best hyper-parameters found for each model by repeatedly performing training and validation for the 250 ms forecast.

| Model | Nr. of layers | Dropout rate | Learning rate | Batch size | L2 weight | Input window length |
|---------------------|---------------|--------------|--------------------|------------|--------------------|---------------------|
| Offline LSTM | 3 | 0 | 5×10^{-4} | 64 | 0 | 32 |
| Offline+online LSTM | 3 | 0 | 1×10^{-6} | Fixed | 1×10^{-6} | 32 |
| Offline LR | — | — | — | — | 1×10^{-2} | 32 |
| Online LR | — | — | — | — | 1×10^{-5} | 16 |

Table A2. Best hyper-parameters found for each model by repeatedly performing training and validation for the 500 ms forecast.

| Model | Nr. of layers | Dropout rate | Learning rate | Batch size | L2 weight | Input window length |
|---------------------|---------------|--------------|--------------------|------------|--------------------|---------------------|
| Offline LSTM | 5 | 0 | 5×10^{-4} | 128 | 0 | 32 |
| Offline+online LSTM | 5 | 0 | 1×10^{-6} | Fixed | 1×10^{-6} | 32 |
| Offline LR | — | — | — | — | 1×10^{-2} | 32 |
| Online LR | — | — | — | — | 1×10^{-4} | 8 |

Table A3. Best hyper-parameters found for each model by repeatedly performing training and validation for the 750 ms forecast.

| Model | Nr. of layers | Dropout rate | Learning rate | Batch size | L2 weight | Input window length |
|---------------------|---------------|--------------|--------------------|------------|--------------------|---------------------|
| Offline LSTM | 5 | 0 | 5×10^{-4} | 64 | 0 | 32 |
| Offline+online LSTM | 5 | 0 | 1×10^{-6} | Fixed | 1×10^{-6} | 32 |
| Offline LR | — | — | — | — | 1×10^{-2} | 32 |
| Online LR | — | — | — | — | 1×10^{-5} | 8 |


Table A4. Mean and standard deviation of MEs for the LMU testing set without breath-holds. The ME of the best performing model is shown in bold for each forecasted time span.

| Model | 250 ms forecast | 500 ms forecast ME [mm] | 750 ms forecast |
|---------------------|--------------------|-------------------------|--------------------|
| Offline LSTM | 2.30 ± 2.04 | 5.12 ± 3.99 | 8.19 ± 5.57 |
| Offline+online LSTM | 2.18 ± 1.95 | 4.81 ± 3.73 | 7.81 ± 5.21 |
| Offline LR | 2.43 ± 2.09 | 5.31 ± 4.15 | 8.33 ± 5.87 |
| Online LR | 2.83 ± 2.05 | 6.02 ± 3.78 | 9.88 ± 6.81 |

ORCID iDs

Elia Lombardo  <https://orcid.org/0000-0003-4522-8298>

Moritz Rabe  <https://orcid.org/0000-0002-7085-4066>

Davide Cusumano  <https://orcid.org/0000-0001-8626-0003>

Marco Riboldi  <https://orcid.org/0000-0002-2431-4966>

Guillaume Landry  <https://orcid.org/0000-0003-1707-4068>

References

- Corradini S *et al* 2019 MR-guidance in clinical reality: current treatment challenges and future perspectives *Radiat. Oncol.* **14** 92
- Cusumano D *et al* 2021 Artificial intelligence in magnetic resonance guided radiotherapy: Medical and physical considerations on state of art and future perspectives *Phys. Med.* **85** 175–91
- Fayad H J *et al* 2012 A generic respiratory motion model based on 4D MRI imaging and 2D image navigators *IEEE Nuclear Science Symp. and Medical Imaging Conf.*

- Friedman M 1937 The use of ranks to avoid the assumption of normality implicit in the analysis of variance *J. Am. Stat. Assoc.* **32** 675–701
- Ge Y et al 2014 Toward the development of intrafraction tumor deformation tracking using a dynamic multi-leaf collimator *Med. Phys.* **41** 061703
- Glitzner M et al 2019 Technical note: MLC-tracking performance on the Elekta Unity MRI-linac *Phys. Med. Biol.* **64** 15NT02
- Green O et al 2018 First clinical implementation of real-time, real anatomy tracking and radiation beam control *Med. Phys.* **45** 3728–40
- Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
- Jackson S et al 2019 MRI B0 homogeneity and geometric distortion with continuous linac gantry rotation on an Elekta Unity MR-linac *Phys. Med. Biol.* **64** 12NT01
- Joehl A et al 2020 Performance comparison of prediction filters for respiratory motion tracking in radiotherapy *Med. Phys.* **47** 643–50
- Josipovic M, Anzar M C, Thomsen J B, Scherman J, Damkjaer S M, Nygard L, Specht L, Pohl M and Persson G F 2019 Deep inspiration breath hold in locally advanced lung cancer radiotherapy: validation of intrafractional geometric uncertainties in the INHALE trial *Br J Radiol.* **92** 1104
- Keall P et al 2021 AAPM task group 264: the safe clinical implementation of MLC tracking in radiotherapy *Med. Phys.* **48** e44–64
- Kim T et al 2020 Clinical experience of MRI4D QUASAR motion phantom for latency measurements in 0.35 T MR-Linac *J. Appl. Clin. Med. Phys.* **22** 128–36
- Kingma D P and Ba J 2015 Adam: A method for stochastic optimization *3rd Int. Conf. for Learning Representations*
- Klueter S 2019 Technical design and concept of a 0.35 T MR-linac *Clin. Transl. Radiat. Oncol.* **18** 98–101
- Krauss A, Nill S and Oelfke U 2011 The comparative performance of four respiratory motion predictors for real-time tumour tracking *Phys. Med. Biol.* **56** 5303–17
- Kurz C et al 2020 Medical physics challenges in clinical MR-guided radiotherapy *Radiat. Oncol.* **15** 93
- Lin H et al 2019 Towards real-time respiratory motion prediction based on long short-term memory neural networks *Phys. Med. Biol.* **64** 085010
- Liney G P et al 2018 MRI-linear accelerator radiotherapy systems *Clin. Oncol.* **30** 686–91
- Menten M J et al 2016 Lung stereotactic body radiotherapy with an MR-linac—quantifying the impact of the magnetic field and real-time tumor tracking *Radiother. Oncol.* **119** 461–6
- Menten M J et al 2020 Automatic reconstruction of the delivered dose of the day using MR-linac treatment log files and online mr imaging *Radiother. Oncol.* **145** 88–94
- Murphy M J and Pokhrel D 2009 Optimization of an adaptive neural network to predict breathing *Med. Phys.* **36** 40–7
- Nemenyi P B 1963 Distribution-free multiple comparisons *PhD Thesis* Princeton University
- Paganelli C et al 2018a Feasibility study on 3D image reconstruction from 2D orthogonal cine-MRI for MRI-guided radiotherapy *J. Med. Imaging Radiat. Oncol.* **62** 389–400
- Paganelli C et al 2018b MRI-guidance for motion management in external beam radiotherapy: current status and future challenges *Phys. Med. Biol.* **63** 22TR03
- Paszke A et al 2017 Automatic differentiation in PyTorch *NIPS-W*
- Pedregosa F et al 2011 Scikit-learn: Machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30
- Placidi L et al 2020 On-line adaptive MR guided radiotherapy for locally advanced pancreatic cancer: Clinical and dosimetric considerations *Tech. Innov. Patient Support Radiat. Oncol.* **15** 15–21
- Poulsen P R et al 2010 Detailed analysis of latencies in image-based dynamic MLC tracking *Med. Phys.* **37** 4998–5005
- Rabe M et al 2021 Porcine lung phantom-based validation of estimated 4D-MRI using orthogonal cine imaging for low-field MR-Linacs *Phys. Med. Biol.* **66** 055006
- Remy C et al 2021 Potential of a probabilistic framework for target prediction from surrogate respiratory motion during lung radiotherapy *Phys. Med. Biol.* **66** 105002
- Roerdink J B T M and Meijster A 2000 The watershed transform: definitions, algorithms and parallelization strategies *Fundam. Inform.* **41** 187–228
- Seregni M et al 2016 Motion prediction in mri-guided radiotherapy based on interleaved orthogonal cine-MRI *Phys. Med. Biol.* **61** 872–87
- Sharp G et al 2004 Prediction of respiratory tumour motion for real-time image-guided radiotherapy *Phys. Med. Biol.* **49** 425
- Shen C et al 2020 An introduction to deep learning in medical physics: advantages, potential, and challenges *Phys. Med. Biol.* **65** 05TR01
- Stemkens B et al 2016 Image-driven, model-based 3D abdominal motion estimation for MR-guided radiotherapy *Phys. Med. Biol.* **61** 5335
- Sun W et al 2020 Adaptive respiratory signal prediction using dual multi-layer perceptron neural networks *Phys. Med. Biol.* **65** 185005
- Uijtewaal P et al 2021 Dosimetric evaluation of mri-guided multi-leaf collimator tracking and trailing for lung stereotactic body radiation therapy *Med. Phys.* **48** 1520–32
- Wang R et al 2018 A feasibility of respiration prediction based on deep bi-LSTM for real-time tumor tracking *IEEE Access* **6** 51262–8
- Yu S et al 2020 Rapid prediction of respiratory motion based on bidirectional gated recurrent unit network *IEEE Access* **8** 49424–35
- Yun J et al 2012 An artificial neural network (ANN)-based lung-tumor motion predictor for intrafractional MR tumor tracking *Med. Phys.* **39** 4423–33