DeepCardioPlanner: Deep Learning based tool for automatic Cardiac Magnetic Resonance planning

PEDRO OSÓRIO[1*], MARKUS HENNINGSSON[2,3,4], ALBERTO GOMEZ HERRERO[5,6], RITA G. NUNES[1], AND TERESA M. CORREIA[6,7]

[1] *Institute for Systems and Robotics - Lisboa and Department of Bioengineering, Instituto Superior Técnico – Universidade de Lisboa, Lisbon, Portugal*
[2] *Division of Cardiovascular Medicine, Department of Medical and Health Sciences, Linköping University, Linköping, Sweden*
[3] *3Center for Medical Image Science and Visualization (CMIV), Linköping University, Linköping, Sweden*
[4] *MR Physics, Perspectum Ltd, Oxford, United Kingdom*
[5] *Ultromics Ltd, Oxford, United Kingdom*
[6] *School of Biomedical Engineering Imaging Sciences, King's College London, United Kingdom*
[7] *Centre for Marine Sciences - CCMAR, Faro, Portugal*

July 13, 2023

Cardiac Magnetic Resonance (CMR) is a powerful technique which can be used to perform a comprehensive cardiac examination. However, its adoption is often limited to specialised centres, in part due to the need for highly trained operators to perform the complex procedures of determining the 4 standard cardiac planes: 2-, 3-, 4-chamber and short axis views. Tools for automating this planning process have been proposed (e.g., Cardiac Dot), but still require some user input. Recently, Deep Learning (DL) methods have been proposed to achieve automatic cardiac planning. For example, cardiac anatomic landmark regression from 2D images has been used to prescribe the standard CMR view planes with good results, but it requires extensive manual annotation to build a dataset to train such methods. Manual annotation free methods have also been proposed for computed tomography (CT), but predict each view position and orientation separately. A similar approach based on rapidly acquired volumetric images could be applicable to CMR where automated view planning would be even more valuable. Here, we propose a set of four deep convolutional neural network (CNN) models (*DeepCardioPlanner*), each trained via a multi-task learning (MTL) approach, to predict the orientation and position of each cardiac view plane from a rapidly acquired 3D scan. This work focus on providing a comprehensive overview of the main steps taken while building the final tool, highlighting the various challenges that arose and how they were dealt with. In particular, we cover how we dealt with the particularities of our dataset, we experiment with different network architectures, loss weighting strategies and hyperparameter tuning. We tested the ability of DeepCardioPlanner to automatically plan the four cardiac views on clinically acquired patient CMR data. Test set error metrics revealed to be comparable with the literature, inclusively matching the range of values for inter-operator variability.

ACKOWLEDGEMENT

This document was written and made publicly available as an institutional academic requirement and as a part of the evaluation of the MSc thesis in Biomedical Engineering of the author at Instituto Superior Técnico. The work described herein was performed at the Institute for Systems and Robotics (Lisbon, Portugal), during the period March 2022 - May 2023, under the supervision of Prof. Rita Gouveia Nunes. The thesis was co-supervised by Prof. Teresa Matias Correia from the Centro de Ciências do Mar, Universidade do Algarve (Faro, Portugal) and Markus Henningson from the Division of Cardiovascular Medicine, Department of Medical and Health Sciences, Linköping University (Linköping, Sweden).

INTRODUCTION

Cardiovascular diseases (CVDs) are a leading cause of premature death and disability worldwide, with increasing incidence and substantial associated costs. Cardiac Magnetic Resonance (CMR) has emerged as a comprehensive imaging technique for evaluating and assessing CVDs, offering a wide range of cardiac examination capabilities. However, CMR requires highly trained operators for determining the standard double-oblique view planes: short axis (SAX), 2-chamber (2CH), 3-chamber (3CH), and 4-chamber (4CH) views. These patient-specific planes are traditionally prescribed through a multistep planning process, requiring several scout scans and manual adjustments, which increase the scan time and workflow complexity.

To address these challenges, there is a need for automated solutions that can assist or even fully automate the planning process, simplifying the protocol, reducing operator burden, and minimizing scan time. Automated planning tools would enable more efficient patient examination, enhance reproducibility, and widen access to CMR.

Tools for automating this planning process have been proposed (e.g., Cardiac Dot1), but still require some user input. Recently, Deep Learning (DL) methods have been proposed to achieve automatic cardiac planning 2,3,4,5. For example, cardiac anatomic landmark regression from 2D images has been used to prescribe the standard CMR view planes with good results,2,4 but it requires extensive manual annotation to build a dataset to train such methods. Manual annotation free methods have also been proposed for computed tomography (CT), but predict each view position and orientation seperately3. A similar approach based on rapidly acquired volumetric images could be applicable to CMR where automated view planning would be even more valuable.

Here, we propose a set of four deep convolutional neural network (CNN) models (DeepCardioPlanner), each trained via a multi-task learning approach, to predict the orientation and position of each cardiac view plane from a rapidly acquired 3D scan. We tested the ability of DeepCardioPlanner to automatically plan the four cardiac views on clinically acquired patient CMR data.

In this thesis, we propose DeepCardioPlanner, a set of four deep convolutional neural network (CNN) models trained using a multi-task learning approach. These models are designed to jointly predict the orientation and position of each cardiac view plane in CMR without user input. By utilising rapidly acquired 3D scans, our models can leverage global context and eliminate the need for extensive landmark annotation.

METHODS

The general outline of the preprocessing steps applied to the dataset, the split performed for the training of the several classifiers presented in this paper and the metrics used is depicted on Figure **??** and will be described with further detail in the next subsections.

Dataset

The dataset used to train our models is comprised of 120 3D CMR scans from different patients labelled with the respective view defining vectors of each of the 4 standard CMR view planes. Each plane is defined by the DICOM image position vector ($\overrightarrow{t}$) **[?]** and the DICOM image orientation vectors $\overrightarrow{o_1}$ and $\overrightarrow{o_2}$ **[?]**. Vectors $\overrightarrow{o_1}$ and $\overrightarrow{o_2}$ are unitary, orthogonal to one another and their cross product defines the normal vector ($\overrightarrow{n}$) to the plane, while vector $\overrightarrow{t}$ give us the location of the origin of the plane, meaning the centre of the first voxel of that slice being transmitted. Similarly to the view planes, the location and orientation of the 3D scans also comes specified by an equivalent set of vectors.

Data were acquired on a 1.5T Philips scanner, using standard clinical protocols and an ECG-triggered volumetric bSSFP sequence with field of view of 440x440x150 mm^3 , voxel size of 3x3x3 mm^3 , compressed SENSE acceleration factor six, and scan time of 10 seconds (assuming 60 bpm heart rate). Adding to that, 51% of the scans were acquired via a LGE-Imaging technique thus including the trace of the contrast. The dataset is quite varied as it was obtained from patients from a wide range of ages, with different pathologies and also by multiple operators.

It is important to note that these data were provided by the University of Linköping and its use in this study was approved by that same institution's ethical committee. Adding to that, every patient whose data is included in this dataset is aware of its use and provided their written informed consent.

To evaluate the models on unseen data, the dataset was split into three subsets: 88 samples for training (74%), 16 samples for validation (13%), and 16 samples for testing (13%), following a stratified approach to maintain a similar percentage of volumes with contrast in each subset as in the entire dataset, ensuring unbiased evaluation of the trained models from the same distribution.

Pre-processing

The reference frame with relation to which all the orientations and positions come defined is called DICOM's Reference Coordinate System (RCS), but, since we want our models to be predicting these view planes from the volume, we must ensure that the predictions can be made using a coordinate system centred on the volume. With that in mind, an extra step of transforming each view vector from the aforementioned RCS into a new reference frame centred on the origin of the 3D scan was taken. For that purpose, affine transform matrices were built from the orientation and position vectors of each volume which were then used to perform this transform to the corresponding view vectors.

The nature of our problem allows two equally correct solutions for each plane's orientation, i.e. the one actually prescribed and its flipped version. Consequently, each view's dataset plane orientation distribution can appear bimodal, with a 180°angle between the two modes. A dataset preprocessing step is proposed to transform the label vectors from one mode to the other, thereby ensuring that this non-anatomical variance is not a confounding factor during training.

Image intensity was standardised to improve model convergence. The volume's axis were swapped to ensure label to volume correspondence. The volumes were resized from an isotropic resolution of 3mm to 4mm to reduce computational burden. Data augmentation consisting of additive noise, scaling. translation, rotation, among other transforms, is also used to increase the models' generalisability.

Figure 3 depicts the full preprocessing pipeline applied to a batch of images every training step.

Multi-objective Loss Function

To address the plane position and orientation subtasks simultaneously, training was performed by combining two losses. Leveraging knowledge that a plane is defined by a point within

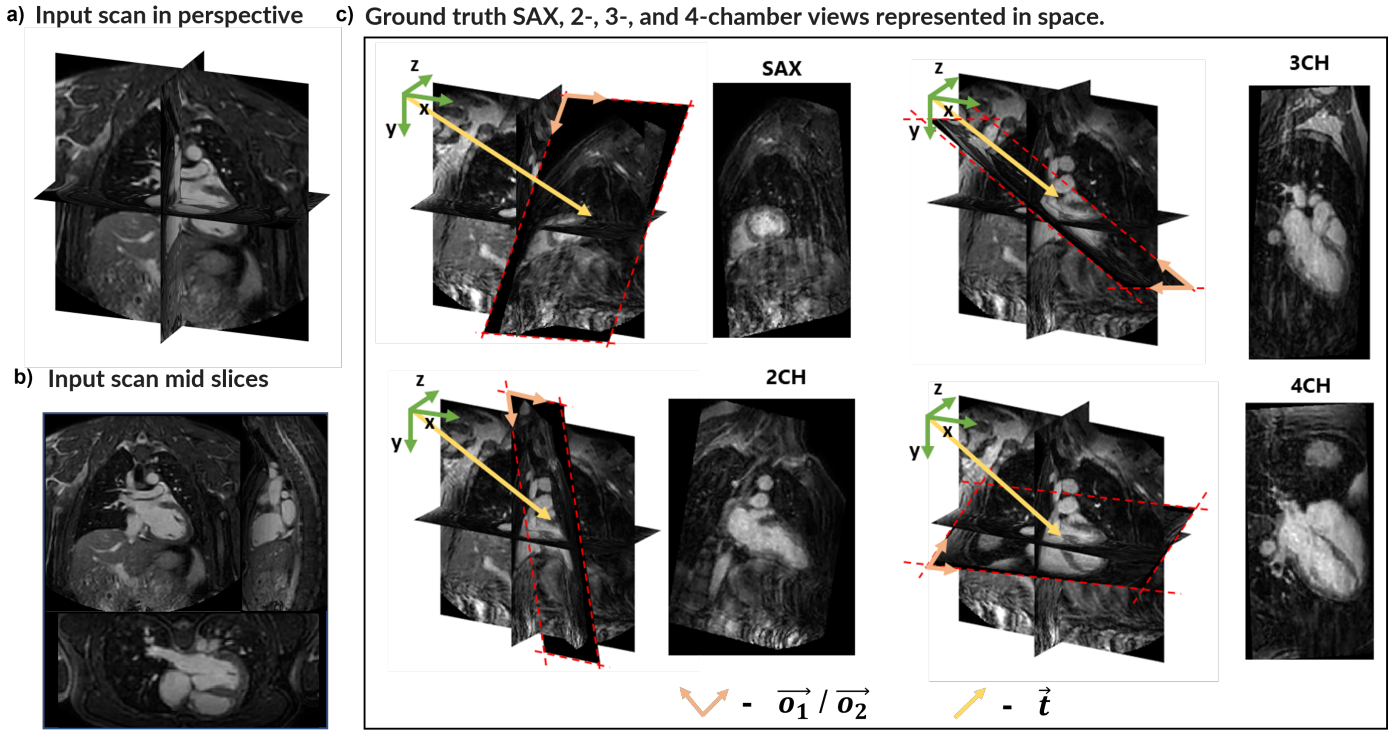

Fig. 1. a) Input volume for a representative subject, b) 2D mid slices (coronal, sagittal and transverse) and c) corresponding ground truth standard view planes and their view defining vectors: $\overrightarrow{o_1}$ and $\overrightarrow{o_2}$ define orientation and $\overrightarrow{t}$ defines the location of the plane's centre. Includes a scheme of each image plane position and orientation within the volume

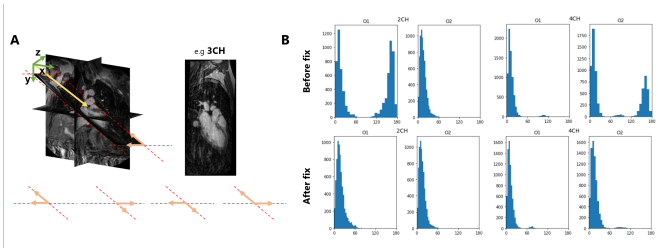

Fig. 2. A) Different orientation vector pairs that depict the same plane orientation. **B)** Distribution of the angles between all 2CH and 4CH $\overrightarrow{o_1}$ and $\overrightarrow{o_2}$ before and after adjusting the dataset.

it and a normal vector, the plane position loss is set to the Euclidean distance between a predicted point and the plane: $\mathcal{L}_{pos} = ||\overrightarrow{d_p}|| = |\overrightarrow{d_c} \cdot \overrightarrow{n}|$, with $\overrightarrow{d_c} = \overrightarrow{t} - \overrightarrow{t}_{pred}$

The orientation loss is computed as the cosine similarity between the predicted and ground truth normal vectors. In fact, two variations of the orientation loss were experimented with. One considers only the exact ground truth $\overrightarrow{n}$ and a viable solution, while the other considers both $\overrightarrow{n}$ and $-\overrightarrow{n}$ as optimal solutions: $\mathcal{L}_{ori} = -cos(\theta)$ and $\mathcal{L}_{ori_{mod}} = 1 - |cos(\theta)|$, with θ angle between $\overrightarrow{n}_{pred}$ and $\overrightarrow{n}$

The two task specific losses need to be combined into a single one that is able to be backpropagated through the shared network. That is achieved via a weighted sum: $\mathcal{L}(W) = w_{pos} \cdot \mathcal{L}_{pos}(W) + w_{ori} \cdot \mathcal{L}_{ori}(W)$, where w_{pos} is the position loss weight and w_{ori} the orientation one.

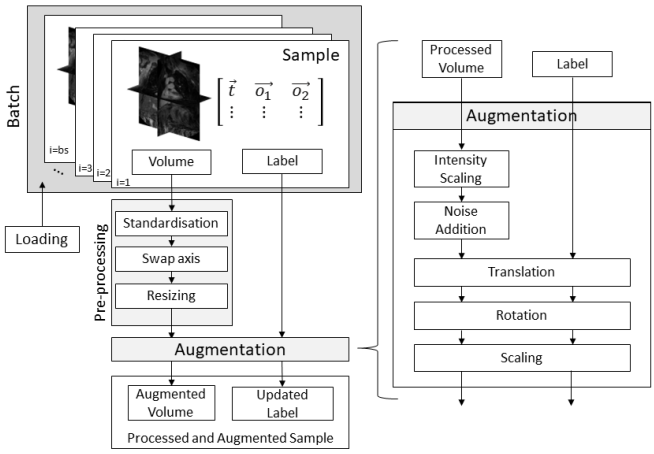

Fig. 3. Loading, pre-processing and augmentation pipeline. All the mentioned operations, from the loading to augmentation, are performed to every sample in a batch during training before feeding them into the network.

A uncertainty based loss weighting approach is also experimented with, where the loss weights were included as learnable parameters of the network through homoscedastic uncertainty [**?**]: $\mathcal{L}(W, \sigma_{pos}, \sigma_{ori}) = \frac{1}{2\sigma_{pos}^2} \cdot \mathcal{L}_{pos}(W) + \frac{1}{2\sigma_{ori}^2} \cdot \mathcal{L}_{ori}(W) + \log(\sigma_{pos}\sigma_{ori})$

Network Architecture

Throughout this project several architectures were tested, but our main results were obtained using two different hard parameter sharing MTL approaches with distinct levels of information sharing: A fully shared architecture (N_A), in which every layer is shared between the two tasks; A multi-headed architecture (N_B) with 2 task-specific heads for separate prediction of the plane position and orientation. In both networks the feature extracting block is shared between the tasks.

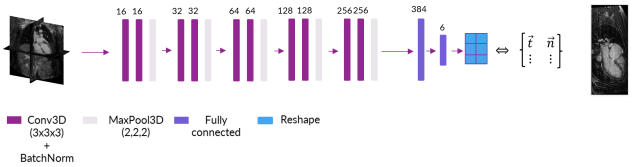

Fig. 4. Baseline model architecture. Numbers on top of the convolutional and dense layers depict the number of filters and units, respectively.

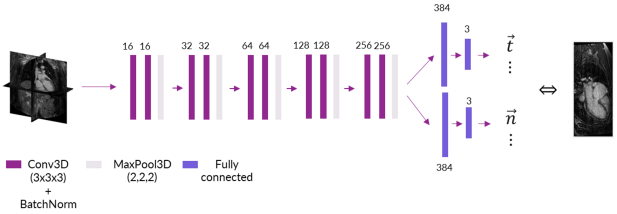

Fig. 5. Architecture with 2 task specific regression heads.

Performance Metrics

Performance on the position prediction sub task is assessed by the displacement error (ε_d) which, just like the position loss, is computed as the distance between the predicted point ($\vec{t}_{pred}$) and the plane in millimetres.

$$\varepsilon_d = ||\vec{d_p}|| = |\vec{d_c} \cdot \vec{n}|, \text{ with } \vec{d_c} = \vec{t} - \vec{t}_{pred} \quad \textbf{(1)}$$

The orientation sub task performance is assessed through the angulation error (ε_θ), which is the angle between the predicted normal vector to the plane ($\vec{n}_{pred}$) and the line defined by the ground truth plane normal vector ($\vec{n}$) in degrees. In practice, this metric can be defined as function of the angle θ between the predicted and true normal vectors. Just like the loss, this metric also considers both $\vec{n}$ and $-\vec{n}$ as equally correct solutions.

$$\varepsilon_\theta = \begin{cases} \theta & \text{for } \theta \leq 90 \\ 180 - \theta & \text{for } \theta > 90 \end{cases} \quad \textbf{(2)}$$

Training Details

As an optimiser we used ADAM with its default parameters as defined in its Keras implementation. The learning rate was set to 10^{-3}. Weight decay was also added to the loss using a weight of 1e-6. Batch size was fixed to 16 samples as more than that would not fit in our GPU's RAM and less would lead to a too unstable training where the gradients estimated are more likely to not accurately represent the overall dataset.

The validation loss was monitored during training with the early stopping callback [**?**] parameterised with a 50 epoch patience, which ensures the training stops after 50 epochs with no validation loss decrease. Also, to ensure the reproducibility of our findings, a *seeding* function was also implemented and ran before each training.

Given the chosen batch size and preprocessing pipeline, training for one epoch (N = 88 samples) takes approximately 40 seconds. Inference times are under 1 second, excluding volume reslicing.

Experimental Setup

The experiments were conducted using Python version 3.10.12 and Keras (Keras: The Python Deep Learning Library https://keras.io/) with TensorFlow (https://www.tensorflow.org/) backend (version 2.12.0). All models were independently trained on a Google Colab Pro instance used was equipped with a 15.35GB Tesla T4 GPU. The complete code base for the experiments is available open source at the following GitHub repository: [https://github.com/pedr0sorio/DeepCardioPlanner].

RESULTS

Dealing with Dataset Challenges

First set of experiment described in this work are aimed at addressing the irregular learning behaviour of the orientation prediction task for the 2CH and 4CH view models, caused by the bimodal distributions of plane orientation in those datasets.

Both the previously described label adjustment preprocessing step, along with the modified cosine similarity orientation loss function are introduced to answer this problem. Both approaches successfully address the problem leading to improved convergence and better performance metrics. A new baseline is proposed using both approaches simultaneously.

Single Branch Network - Automated 2CH view prediction optimisation

The previous baseline shows a notable bias on the orientation task, as none of the models are able to reach very low values, thereby showing evidences of underfitting (see Figure 6). Using the 2CH view prediction as a representative task, the next set of experiments are aimed at increasing the effective capacity of the model in order to escape underfitting. In particular, we will analyse model complexity, experiment with different loss weighting strategies and tune the learning rate.

We find that N_A is complex enough to model each sub task separately. This motivates the search for a better loss weigh combination, finding that increasing the value of w_{ori} successfully compensates for the difference in scales of the two task specific losses, ensuring neither dominates the joint loss and, thus, the training. The learning rate is also tuned.

Here, uncertainty-based loss weighting was introduced. However, the approach showed minimal impact on task learning as the weights remained largely unchanged. This suggests

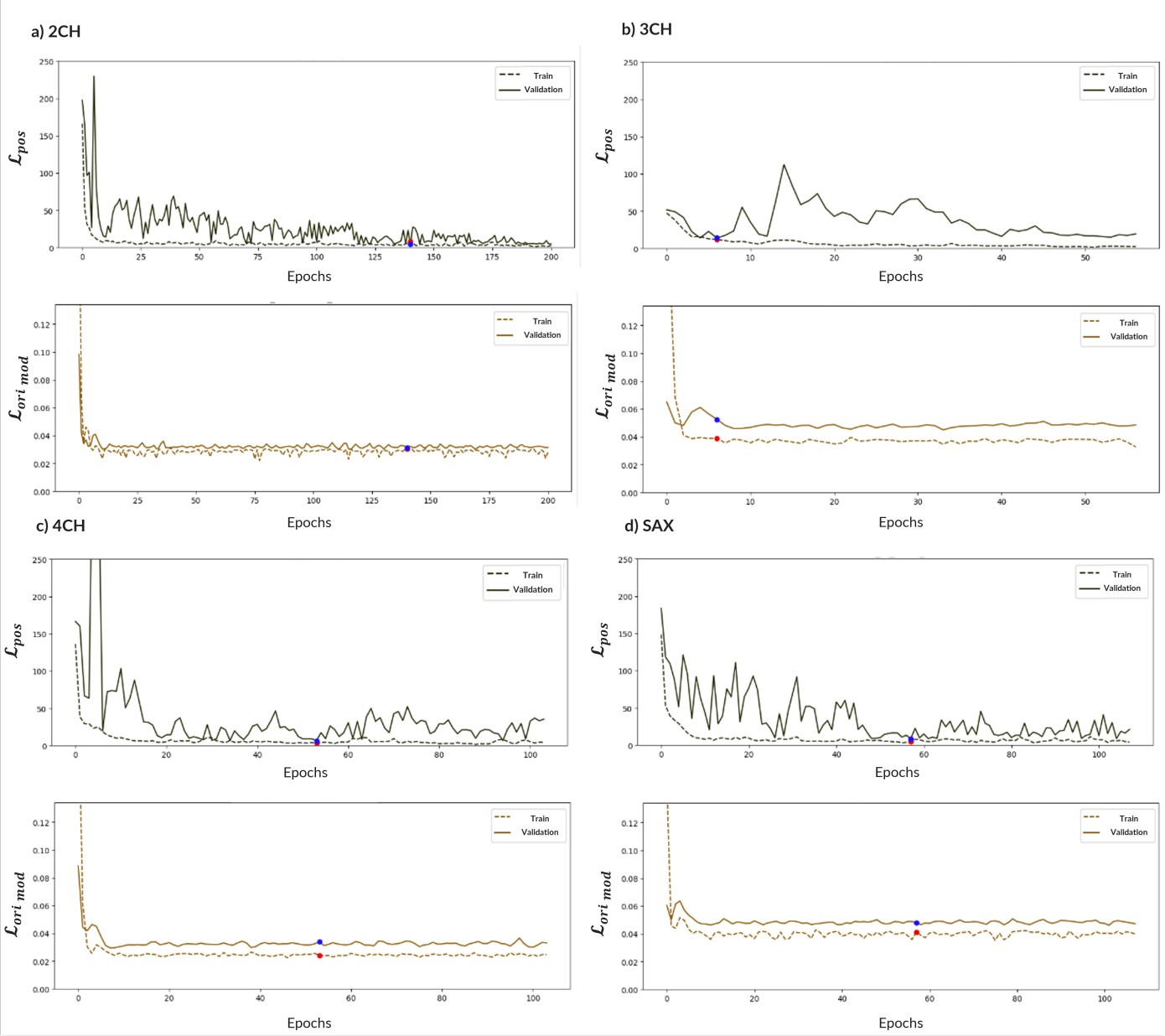

Fig. 6. Training curves for all view models after changing orientation loss to $\mathcal{L}_{ori_{mod}}$ and applying the label adjustment. Best epoch is marked with a blue and red dot on the validation and training curves, respectively.

that the homeoscedastic uncertainty, on which the weighting relies, did not differ significantly between the tasks, limiting its effectiveness.

Multi-Headed Network - Automated 2CH view prediction

After finding an optimal loss weighting setting and learning rate, we proceed with evaluating how different levels of information sharing can impact the model performance and learning behaviour. Architecture N_B is introduced, finding that the level of hard parameter sharing in N_A was excessively constraining the hypothesis space of the model. The new architecture leads to the model reaching not only lower training, but also validation and test set errors (see Figure 7). With this new best training

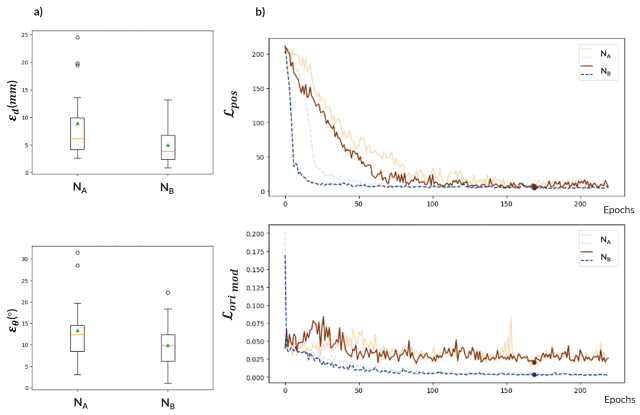

Fig. 7. 2CH view model test set performance (a) and curves (b) when training on a fully shared architecture (N_A) or a multi-headed one (N_B). Validation curve plotted as a full line and the train curve as a dotted one. Best epoch is marked with a blue and red dot on the validation and training curves, respectively.

setting, the effect of data augmentation and volume resolution are evaluated. We demonstrate how data augmentation must be applied moderately to ensure the reality of the transformations. A higher resolution volume did not improve performance, hinting at the fact that finer details might not be as relevant for CMR view prescription as coarser features of the scans.

Finally, one last comparison between MTL and STL is made. Given the design of the losses and the training setting, using a MTL approach outperforms having a separate model for each task (STL). Also, training on solely the position loss leads to a gradual improvement of orientation loss during training which highlights the inter-dependency of the two tasks and explains the better performance achieved when training on both objectives simultaneously.

DeepCardioPlanner - Automated 2CH, 4CH, 3CH and SAX view prediction

Here, we assess the generalisation of the aforementioned changes based on the 2CH view model to the remaining models. We found they generalise satisfactorily, highlighting how all the view predicting models suffered from the same type of underfitting problems we aimed to address for the 2CH model.

DeepCardioPlanner comprises the top-performing models, one for each view, which collectively enable the automatic view prescription for all four standard CMR view planes from a rapidly acquired scan.

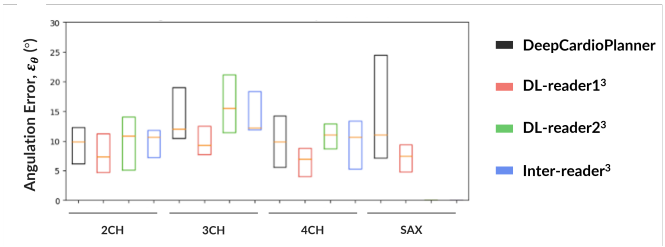

Fig. 8. Comparison of test set angulation error distribution between our tool and the method by Chen et al [**?**]. DeepCardioPlanner: Our performances; DL-reader1: Their performances on samples from the same operator that generated their training set; DL-reader2: Their performances on samples from a never seen operator; Inter-reader: Differences in plane orientation between the two operators.

We further compared *DeepCardioPlanner* with the literature finding that it yields performances metrics within the literature ranges for the same task with CT and wherein STL approach was taken [**?**]. Not only that but the attained error distributions are much alike the inter operator differences from the same work [**?**]. The resemblance in the error distributions further supports the validity and reliability of our model's predictions, as they align with the inherent variability observed in human assessments.

DeepCardioPlanner does not perform as well as other landmark based regression methods, but it achieves promising results considering the limitations of our dataset.

CONCLUSION

In this work we have explored how it is possible to leverage a MTL approach to train one deep learning model to predict each one of the standard SAX, 2CH, 3CH and 4CH CMR view planes, from a rapidly acquired 3D scan (∼10sec). The proposed tool shows potential to greatly reduce examination time and complexity, as it provides an accurate and fully automatic alternative to conventional CMR view planning.

We acknowledge certain limitations inherent to our research design. A larger and more diverse dataset would have helped with the assessment of the generalisation ability of the models. K-folds cross validation could have been used to increase robustness of the results. This did not happen due to the lengthy nature of our trainings and the fact that our access to a GPU was very limited. Also, it is likely that tailoring the training setting to each specific view predicting model would result in improved overall performances. The access to the GPU was also a constraint in this regard.

As future work we leave experimenting with different CNN architectures like ones that exploit residual learning. More refined MTL approaches should also be considered for future developments in this topic.

Also going forward, it would be recommended to perform a statistical analysis in order to assess if the performance error metrics are statistically different from zero. Also, designing and conducting a clinical reading where domain experts would classify the quality and clinical relevance of the predicted planes would contribute with extra robustness to any future results.

Finally, one other interesting avenue to explore is whether we can extend this multi-task learning approach from single view prediction per model to a multi view prediction per model. Training the model to prescribe all view planes jointly might

produce good results by leveraging the relative position and orientation between views.