Deep learning for online adaptive radiotherapy

Elmahdy, M.S.E.

5

Joint Registration and Segmentation via Multi-Task Learning for Adaptive Radiotherapy of Prostate Cancer

This chapter was adapted from:

M Elmahdy, L Beljaards, S Yousefi, H Sokooti, F Verbeek, U van der Heide, and M Staring. **Joint Registration and Segmentation via Multi-Task Learning for Adaptive Radiotherapy of Prostate Cancer**, *IEEE Access*, 2021.

Abstract

Medical image registration and segmentation are two of the most frequent tasks in medical image analysis. As these tasks are complementary and correlated, it would be beneficial to apply them simultaneously in a joint manner. In this paper, we formulate registration and segmentation as a joint problem via a Multi-Task Learning (MTL) setting, allowing these tasks to leverage their strengths and mitigate their weaknesses through the sharing of beneficial information. We propose to merge these tasks not only on the loss level, but on the architectural level as well. We studied this approach in the context of adaptive image-guided radiotherapy for prostate cancer, where planning and follow-up CT images as well as their corresponding contours are available for training. At testing time the contours of the follow-up scans are not available, which is a common scenario in adaptive radiotherapy. The study involves two datasets from different manufacturers and institutes. The first dataset was divided into training (12 patients) and validation (6 patients), and was used to optimize and validate the methodology, while the second dataset (14 patients) was used as an independent test set. We carried out an extensive quantitative comparison between the quality of the automatically generated contours from different network architectures as well as loss weighting methods. Moreover, we evaluated the quality of the generated deformation vector field (DVF). We show that MTL algorithms outperform their Single-Task Learning (STL) counterparts and achieve better generalization on the independent test set. The best algorithm achieved a mean surface distance of 1.06 ± 0.3 mm, 1.27 ± 0.4 mm, 0.91 ± 0.4 mm, and 1.76 ± 0.8 mm on the validation set for the prostate, seminal vesicles, bladder, and rectum, respectively. The high accuracy of the proposed method combined with the fast inference speed, makes it a promising method for automatic re-contouring of follow-up scans for adaptive radiotherapy, potentially reducing treatment related complications and therefore improving patients quality-of-life after treatment. The source code is available at https://github.com/moelmahdy/JRS-MTL.

5.1 Introduction

Medical image analysis aims to extract clinically useful information that aids the diagnosis, prognosis, monitoring and treatment of diseases [92, 93]. Two of the most common tasks in such analyses are image registration and segmentation [94]. Image segmentation aims to identify and cluster objects that prevail similar characteristics into distinctive labels, where these labels can be used for diagnosis or treatment planning. Image registration is the task of finding the geometrical correspondence between images that were acquired at different time steps or from different imaging modalities. These two tasks are complementary, as for example image atlases warped by image registration algorithms are often used for image segmentation [21, 22], while image contours can be used to guide the image registration method in addition to the intensity images [23, 17, 24]. Contours are also used for evaluating the quality of the registration [25, 26]. However, each of these tasks has its own strengths and weaknesses. For instance, image segmentation algorithms can directly delineate images based on texture and surrounding anatomy, and may therefore be robust to large organ deformations. However it sometimes has difficulties with low contrast areas and irregularly shaped organs. On the other hand, image registration algorithms have the ability to encode prior knowledge of the patient's anatomy and therefore may perform better on low quality images. However, such methods sometimes have difficulty with large deformations. Therefore, coupling of image registration and segmentation tasks and modeling them in a single network could leverage their strengths and mitigate their weaknesses through the sharing of beneficial information.

Adaptive image-guided radiotherapy is an exemplar application where the coupling of image registration and segmentation is vital. In radiotherapy, treatment radiation dose is delivered over a course of multiple inter-fraction sessions. In an adaptive setting, re-imaging of the daily anatomy and automatic re-contouring is crucial to compensate for patient misalignment, to compensate for anatomical variations in organ shape and position, and an enabler for the reduction of treatment margins or robustness settings [95, 96]. These have an important influence on the accuracy of the dose delivery, and improve the treatment quality, potentially reducing treatment related side-effects and increasing quality-of-life after treatment [97]. Automatic contouring can be done by direct segmentation of the daily scan, or by registration of the annotated planning scan with the daily scan followed by contour propagation. Image registration has the advantage of leveraging prior knowledge from the initial planning CT scan and the corresponding clinical-quality delineations, which may especially be helpful for challenging organs. On the other hand, image segmentation methods may better delineate organs that vary substantially in shape and volume between treatment fractions, which is often the case for the rectum and the bladder.

In this study, we propose to fuse these tasks at the network architecture level as well as via the loss function. Our key contributions in this paper are as follows:

1. We formulate image registration and segmentation as a multi-task learning problem, which we explore in the context of adaptive image-guided radiotherapy.

2. We explore different joint network architectures as well as loss weighting methods for merging these tasks.

3. We adopt the cross-stitch network architecture for segmentation and registration tasks and explore how these cross-stitch units facilitate information flow between these tasks.

4. Furthermore, we compare MTL algorithms against single-task networks. We demonstrate that MTL algorithms outperform STL networks for both segmentation and registration tasks. To the best of our knowledge this is the first study to investigate various MTL algorithms on an architectural level as well as on a loss weighing level for joint registration and segmentation tasks.

5. We thoroughly investigate the internals of the STL and MTL networks and pinpoint the best strategy to merge this information to maximize the information flow between the two tasks.

Initial results of this work were presented in [98], focusing on the cross-stitch unit in a proposed joint architecture. In the current paper we extend this study to the architectural fusion of these tasks as well as different loss weighting mechanisms. Moreover, an extensive analysis of the different methodologies was performed, detailing the effect of architectural choices, information flow between the two tasks, etc.

The remainder of this paper is organized as follows: Section 5.2 introduces single-task networks, multi-task networks, and loss weighting approaches. In Section 5.3 we introduce the datasets and details about the implementation as well as the experiments. In Sections 5.5 and 5.6, we discuss our results, provide future research directions, and present our conclusions.

5.1.1 Related work

In the last decade, researchers have been exploring the idea of fusing image segmentation and registration. Lu *et al.* [99] and Pohl *et al.* [100] proposed modeling these tasks using a Bayesian framework such that these tasks would constrain each other. Yezzi [101] proposed to fuse these tasks using active contours, while Unal *et al.* [102] proposed to generalize the previous approach by using partial differential equations without shape priors. Mahapatra *et al.* [24] proposed a Joint Registration and Segmentation (JRS) framework for cardiac perfusion images, where the temporal

intensity images are decomposed into sparse and low rank components corresponding to the intensity change from the contrast agent and the motion, respectively. They proposed to use the sparse component for segmentation and the low rank component for registration. However, most of the aforementioned methods require complex parameter tuning and yield long computation times.

Recently, deep learning-based networks have shown unprecedented success in many fields especially in the medical image analysis domain [20, 103, 104, 105, 106, 13], where deep learning models perform on par with medical experts or even surpassing them in some tasks [107, 108, 109, 110]. Several deep learning-based approaches have been proposed for joint registration and segmentation. The joining mechanisms in the literature can be classified in two categories, namely joining via the loss function and via the architecture as well as the loss function. Selected exemplar methods of the first approach are Hue *et al.* [111], who proposed to join segmentation and registration via a multi-resolution Dice loss function. Elmahdy *et al.* [23] proposed a framework that is a hybrid between learning and iterative approaches, where a CNN network segments the bladder and feeds it to an iterative-based registration algorithm. The authors integrated domain-specific knowledge such as air pocket inpainting as well as contrast clipping, moreover they added an extra registration step in order to focus on the seminal vesicles and rectum. Elmahdy *et al.* [17] and Mahapatra *et al.* [112] proposed a GAN-based (Generative Adversarial Network) approach, where a generative network predicts the correspondence between a pair of images and a discriminator network for giving feedback on the quality of the deformed contours. Exemplar methods of the second category are Xu *et al.* [113], who presented a framework that simultaneously trains a registration and a segmentation network. The authors proposed to jointly learn these tasks during training, however the networks can be used independently during test time. This enables prediction of only the registration output, when the labels are not available during test time. Estienne *et al.* [114] proposed to merge affine and deformable registration as well as segmentation in a 3D end-to-end CNN network. Recently Liu *et al.* [115] proposed an end-to-end framework called JSSR that registers and segments multi-modal images. This framework is composed of three networks: a generator network, that synthesizes the moving image to match the modality of the fixed image, a registration network that registers the synthesized image to the fixed image, and finally a segmentation network that segments the fixed, moving, and synthesized images.

All the previous methods explored the idea of joining segmentation and registration, where to the best of our knowledge none have explored how these tasks are best connected and how to optimize the information flow between them on both the loss and architectural levels.

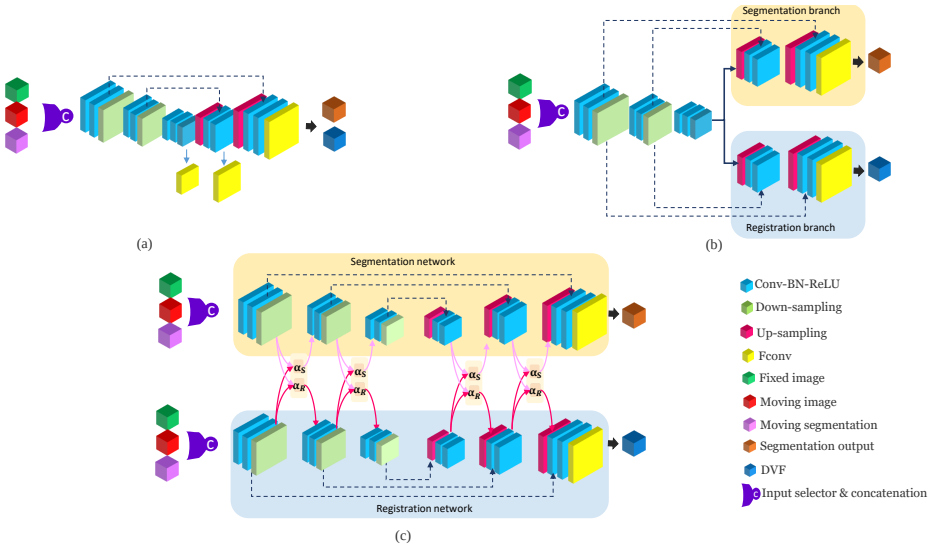

Figure 5.1: The proposed network architectures introduced in the paper. (a) is the base STL network architecture for either segmentation or registration, but also represents the dense parameter sharing MTL network architecture; (b) is the architecture with a shared encoder, while (c) is the Cross-stitch network architecture. Details about the number of feature maps are presented in Section 5.3.2.

5.2 Methods

5.2.1 Base network architecture

The base architecture for the networks in this paper is a 3D CNN network inspired by the U-Net and BIRNet architectures [116, 117]. Figure 5.1a shows the architecture of the base network. The network encodes the input through $3 \times 3 \times 3$ convolution layers with no padding. LeakyReLU [118] and batch normalization [119] are applied after each convolutional layer. We used strided convolutions in the down-sampling path and trilinear upsampling layers in the upsampling path. Through the upsampling path, the number of feature maps increases while the size of the feature maps decreases, and vice versa for the down-sampling path. The network has three output resolutions and is deeply supervised at each resolution. Each resolution is preceded by a $1 \times 1 \times 1$ fully convolution layer (Fconv) so that at coarse resolution, the network can focus on large organs as well as large deformations, while vice versa at fine resolution. In order to extract the groundtruth for different resolutions, we perform cropping of different sizes as well as strided sampling so that for every input patch of size n^3, the sizes of the coarse, mid, and fine resolution are $(\frac{n}{4} - 7)^3$, $(\frac{n}{2} - 18)^3$, and $(n - 40)^3$, respectively.

5.2.2 Single task learning

Single-task networks are designed to solve one task and therefore require a large amount of labeled training samples, which are scarce in the medical domain since it takes time and trained medical personnel to contour these images. The segmentation and registration networks have the same architecture as the base network depicted in Figure 5.1a, but differ in the input and output layers. Here, single-task networks are considered baseline networks for comparing with the performance of the proposed multi-task networks.

5.2.2.1 Segmentation network

The input to the segmentation network is the daily CT scan, referred to as the fixed image I_f, where the network predicts the corresponding segmentation S_f^{pred}. S_f^{pred} represents the probability maps for the background, target organs, and organs-at-risk. The network was trained using the Dice Similarity Coefficient (DSC) loss, which quantifies the overlap between the network prediction S_f^{pred} and the groundtruth S_f as follows:

$$\mathscr{L}_{\text{DSC}} = 1 - \frac{1}{K}\sum_{k=1}^{K}\frac{2 * \sum_x S_k^{\text{pred}}(x) \cdot S_k(x)}{\sum_x S_k^{\text{pred}}(x) + \sum_x S_k(x)}, \tag{5.1}$$

where K is the number of structures to be segmented, x is the voxel coordinate, S_k is the ground truth segmentation, and S_k^{pred} the predicted probabilities. The network has 779,436 trainable parameters.

5.2.2.2 Registration network

The input to the registration network is the concatenation of the planning scan, referred to as the moving image $I_m$and the daily scan I_f. The network predicts the geometrical correspondence between the input images. This correspondence is represented by the displacement vector field (DVF), referred to as ϕ^{pred}. This DVF is then used to warp I_m. In an ideal scenario, the warped moving image I_m^{warped} would be identical to I_f. The network is trained using Normalized Cross Correlation (NCC) in order to quantify the dissimilarity between I_m^{warped} and I_f. Since the images are from a single imaging modality (CT) with a similar intensity distribution, NCC is an obvious choice abundantly used in the registration literature. Moreover, the implementation is straightforward and efficient when using plain convolution operations. NCC is defined by the following equation:

$$\mathscr{L}_{\text{NCC}} = 1 - \frac{\sum_x [(I_f(x) - \overline{I_f}) \cdot (I_m^{\text{warped}}(x) - \overline{I_m^{\text{warped}}})]}{\sigma_{I_f}\sigma_{I_m^{\text{warped}}}}, \tag{5.2}$$

where x is the voxel coordinate, and σ_{I_f} and $\sigma_{I_m^{\text{warped}}}$ are the standard deviation of the fixed and warped images, respectively. In order to encourage the network to predict a

smooth DVF, a bending energy penalty term is added for regularization:

$$\mathscr{L}_{\text{BE}} = \frac{1}{N} \sum_{x} \| H(\phi^{pred}(x)) \|_2^2, \tag{5.3}$$

where H is the Hessian matrix. Now the total registration loss becomes:

$$\mathscr{L}_{\text{Registration}} = \mathscr{L}_{\text{NCC}} + w \cdot \mathscr{L}_{\text{BE}}, \tag{5.4}$$

where w is the bending energy weight. For more details on the selection of w, see Section 5.4.1. The network has 779,733 trainable parameters.

5.2.3 Multi task learning

In Multi-Task Learning (MTL), related tasks regularize each other by introducing an inductive bias, thus making the model agnostic to overfitting compared to its STL counterparts [120]. MTL can also be considered as an implicit data augmentation strategy, since it effectively increases the training sample size while encouraging the model to ignore data-dependent noise. Because different tasks have different noise patterns, modeling these tasks simultaneously enables the model to generalize well [121]. Moreover, in MTL models, some features can be more easily learned by one task than another, thus encouraging information cross-talk between tasks [122].

Also, in real-world scenarios, physicians usually incorporate knowledge from different imaging modalities or previous tasks in order to come up with a diagnosis or better understanding of the underlying problem. This illustrates that the knowledge embedded in one task can be leveraged by other tasks and hence it is beneficial to jointly learn related tasks.

Choosing the architecture of an MTL network is based on the following two factors [123]: *what to share* and *how to share*. *What to share* defines the form in which knowledge is shared between tasks. This knowledge sharing can be done through hand-crafted features, input images, and model parameters. *How to share* determines the optimal manner in which this knowledge is shared. In this paper, we focus on parameter-based sharing.

In the following sections, we investigate different MTL network architectures in order to best understand how segmentation and registration tasks share information on the architectural level. The investigated networks predict two sets of contours, one set resulting from the segmentation task and one from the registration task. In this paper, we select the best set of contours as the final output, based on the validation results. More sophisticated strategies are discussed in Section 5.5.

5.2.3.1 Joint registration and segmentation via the registration network

The network in this method, dubbed JRS-reg, has the same architecture as the STL registration network from Section 5.2.2.2, except that this network is optimized using a joint loss as presented in Eq. 5.6.

5.2.3.2 Dense parameter sharing

In this architecture both segmentation and registration tasks are modeled using a single network, where both tasks share all parameters except for the task-specific parameters in the output layer, see Figure 5.1a. The network architecture is the same as the base network (see Section 5.2.1) except for the input and output layers. This dense sharing eliminates overfitting issues since it enforces the parameters to model all the tasks at once, however it does not guarantee the best representation for individual tasks [123]. The input to the network is the concatenation of I_m, I_f, and S_m. The network predicts the ϕ^{pred} between input images as well as S_f^{pred}. The network has 781,164 trainable parameters.

5.2.3.3 Encoder parameter sharing

Since the input to the segmentation and registration tasks are both CT scans, this means they both encode similar features in the down-sampling path of the network. Therefore in this network both tasks share the encoding path and then splits into two upsampling task specific decoder paths. We call this network the Shared Encoder Double Decoder (SEDD) network. Figure 5.1b shows the architecture of the network. The input to the network is the concatenation of I_m, I_f, and S_m. The network predicts ϕ^{pred} between the input images from the registration path while predicting S_f^{pred} from the segmentation path. The network has 722,936 trainable parameters.

5.2.3.4 Cross-stitch network

A flexible approach to share parameters is via a Cross-Stitch (CS) network [124]. In contrast to the heuristic approach of manually choosing which layers are shared and which are task-specific, the CS network introduces a learning-based unit to determine the amount of feature sharing between tasks. The CS units learn to linearly combine feature maps from the two networks, one for segmentation and one for registration, as shown in Figure 5.1c. The unit itself is defined as:

$$\begin{bmatrix} \bar{X}_S^{\ell,k} \\ \bar{X}_R^{\ell,k} \end{bmatrix} = \begin{bmatrix} \alpha_{SS}^{\ell,k} & \alpha_{SR}^{\ell,k} \\ \alpha_{RS}^{\ell,k} & \alpha_{RR}^{\ell,k} \end{bmatrix} \begin{bmatrix} X_S^{\ell,k} \\ X_R^{\ell,k} \end{bmatrix}, \tag{5.5}$$

where $X_S^{\ell,k}$ and $X_R^{\ell,k}$ represent the feature maps k at layer l for the segmentation and registration networks, respectively. $\alpha_{SS}^{\ell,k}$, $\alpha_{SR}^{\ell,k}$, $\alpha_{RS}^{\ell,k}$, and $\alpha_{RR}^{\ell,k}$ represent the learnable parameters of the CS unit. $\bar{X}_S^{\ell,k}$ and $\bar{X}_R^{\ell,k}$ are the output feature maps for the segmentation and registration networks, respectively. The advantage of CS units is that the network can dynamically learn to share the feature maps in case this is beneficial in terms of the final loss value. In case there is no benefit, an identity matrix can be learned, so that the feature maps become task-specific. This allows the network to learn a smooth sharing between the tasks at a negligible increase in the

number of parameters. As suggested by the original paper, we placed the CS units after the downsampling and upsampling layers resulting in a total of 4 CS units. The CS network has 779,000 trainable parameters.

5.2.4 Loss weighting

The loss function for the MTL networks is defined by:

$$\mathscr{L} = w_0 \cdot \mathscr{L}_{\text{NCC}} + w_1 \cdot \mathscr{L}_{\text{DSC-R}} + w_2 \cdot \mathscr{L}_{\text{DSC-S}} + w_3 \cdot \mathscr{L}_{\text{BE}}, \tag{5.6}$$

where w_i are the loss weights. They are chosen based on the relative contribution of their corresponding tasks, so that different tasks would learn at the same pace. These weights can be chosen manually based on empirical knowledge, or automatically. A simple choice would be to weigh the losses equally with a fixed weight of 1. Following are some exemplar algorithms for choosing the loss weights automatically. Chen *et al.* proposed GradNorm [125] to weigh different tasks by dynamic tuning of the gradient magnitudes of the tasks. This tuning is achieved by dynamically changing the learning rate for each task so that all tasks would be learning at the same speed. The drawback of this approach is that it requires access to the internal gradients of the shared layers which could be cumbersome. Moreover, one needs to choose which shared layer to back propagate to in case of multiple shared layers. Kendall *et al.* [126] proposed to weigh each task by considering the homoscedastic uncertainty of that task, so that tasks with high output variance will be weighted less than tasks with low variance. This approach only adds few trainable parameters, namely equal to the number of loss functions. Inspired by GradNorm, Liu *et al.* proposed Dynamic Weight Averaging (DWA) [127], where each task is weighted over time by considering the rate of change of the relative loss weights. Contrary to GradNorm, DWA only requires the numerical values of the loss functions rather than their derivatives. In this paper, we compared equal weights versus homoscedastic uncertainty and DWA. For all the experiments, we set the weight of the bending energy to a fixed value of 0.5 (for more details see Section 5.4.1) instead of a trainable one. This is to prevent the network to set it too low in order to improve the DSC of the deformed contours on the account of the smoothness of the predicted DVF.

5.2.4.1 Homoscedastic uncertainty

Homoscedastic uncertainty was proposed as a loss weighting method by Kendall *et al.* [126]. This is a task-dependant uncertainty which is not dependant on the input data but rather varies between tasks. The authors derived their finding by maximizing the Gaussian likelihood while considering the observational noise scalar σ that represents the homoscedastic uncertainty term related to each task. The following equation describes the weight loss using homoscedastic uncertainty, where σ is a trainable

parameter:

$$\mathscr{L}_{\text{homoscedastic}} = \sum_{i=1}^{T} \frac{1}{\sigma_i^2} \mathscr{L}_i + \log \sigma_i, \tag{5.7}$$

where T is the number of tasks. The higher the uncertainty of task i, the lower the contribution of its associated loss $\mathscr{L}_i$ to the overall loss. The log term can be viewed as a regularization term, so that the network would not learn a trivial solution by setting the uncertainty of all tasks to extreme values.

5.2.4.2 Dynamic weight averaging

Dynamic Weight Averaging (DWA) was proposed by Liu *et al.* [127]. Similar to GradNorm [125], DWA weights the losses via the rate of change of the loss of each task over the training iterations t. In contrast to GradNorm, DWA does not require access to the internal gradients of the network, but only requires the numerical loss values. According to DWA, the weight w of the loss $\mathscr{L}$ associated with the task k is defined as:

$$w_k(t) = \frac{K \exp(r_k(t-1)/tmp)}{\sum_i \exp(r_i(t-1)/tmp)}, \; r_k(t-1) = \frac{\mathscr{L}_k(t-1)}{\mathscr{L}_k(t-2)}, \tag{5.8}$$

where r_k is the relative loss ratio and tmp is the temperature that controls the smoothness of the the task weighting. Here, we set $tmp = 1$ as suggested by the original paper. For the initial two iterations, $r_k(t)$ is set to 1.

5.3 Datasets, implementation, and evaluation

5.3.1 Datasets

This study involves two datasets from two different institutes and scanners for patients who underwent intensity-modulated radiotherapy for prostate cancer. The first dataset is from Haukeland Medical Center (HMC), Norway. The dataset has 18 patients with 8-11 daily CT scans, each corresponding to a treatment fraction. These scans were acquired using a GE scanner and have 90 to 180 slices with a voxel size of approximately 0.9 × 0.9 × 2.0 mm. The second dataset is from Erasmus Medical Center (EMC), The Netherlands. This dataset consists of 14 patients with 3 daily CT scans each. The scans were acquired using a Siemens scanner, and have 91 to 218 slices with a voxel size of approximately 0.9 × 0.9 × 1.5 mm. The target structures (prostate and seminal vesicles) as well as organs-at-risk (bladder and rectum) were manually delineated by radiation oncologists. All datasets were resampled to an isotropic voxel size of 1 × 1 × 1 mm. All scans and corresponding contours were affinely registered beforehand using `elastix` [128], so that corresponding anatomical structures would fit in the network's field of view. The scan intensities were clipped to [-1000, 1000] .

5.3.2 Implementation and training details

All experiments were developed using Tensorflow (version 1.14) [129]. The convolutional layers were initialized with a random normal distribution ($\mu = 0.0$, $\sigma = 0.02$). All parameters of the Cross-stitch units were initialized using a truncated normal distribution ($\mu = 0.5$, $\sigma = 0.25$) in order to encourage the network to share information at the beginning of the training. In order to ensure fairness regarding the number of parameters in all the networks, the number of filters for the Cross-stitch network were set to [16, 32, 64, 32, 16], while for the other networks the numbers were scaled by $\sqrt{2}$ resulting in [23, 45, 91, 45, 23] filtermaps. This results in approximately 7.8×10^5 trainable parameters for each network. The networks were trained using the RAdam optimizer [130] with a fixed learning rate of 10^{-4}. Patches were sampled equally from the target organs, organs-at-risk and torso. All networks were trained for 200K iterations using an initial batch size of 2. The batch size is then doubled by switching the fixed and moving patches so that the network would warp the fixed patch to the moving patch and vice versa at the same training iteration.

The networks were trained and optimized on the HMC dataset, while the EMC dataset was used as an independent test set. Training was performed on a subset of 111 image pairs from 12 patients, while validation and optimization was carried out on the remaining 50 image pairs from 6 patients.

From each image, 1,000 patches of size 96 × 96 × 96 voxels were sampled. The size of the patch was chosen so that it would fit in the GPU memory, while still producing a patch size of 17^3 at the lowest resolution, which is a reasonable size to encode the deformation from the surrounding region. Losses from the deeply supervised resolutions were weighted equally, $\frac{1}{3}$ each. Training was performed on a cluster equipped with NVIDIA RTX6000, Tesla V100, and GTX1080 Ti GPUs with 24, 16 and 11 GB of memory, respectively. The source code is available at https://github.com/moelmahdy/JRS-MTL.

5.3.3 Evaluation metrics

The automatically generated contours are evaluated geometrically by comparing them against the manual contours for the prostate, seminal vesicle, rectum, and bladder. The Dice similarity coefficient (DSC) measures the overlap between contours:

$$\mathrm{DSC} = \sum \frac{2 \mid S_f \cap S_g \mid}{\mid S_f \mid + \mid S_g \mid}, \tag{5.9}$$

where S_g is the generated contour from either the segmentation or the registration network. The distance between the contours is measured by the Mean Surface Distance

Table 5.1: The effect of network input for the different architectures on the validation set (HMC) in terms of MSD (mm). Lower values are better. Here, $\oplus$ is the concatenation operation, and $\cdot\|\cdot$ represents the inputs to the segmentation network (left of $\|$) and the inputs to the registration network (right of $\|$). Stars denote one-way ANOVA statistical significance with respect to the Cross-stitch network with $I_f \,||\, I_f \oplus I_m \oplus S_m$ as inputs.

			Prostate		Seminal vesicles		Rectum		Bladder	
Network	Input	Output path	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median
Seg	I_f		1.49 ± 0.3*	1.49	2.50 ± 2.6	2.09	3.39 ± 2.2	2.73	1.60 ± 1.1*	1.13
	$I_f \oplus S_m$		1.31 ± 0.4	1.23	1.63 ± 0.9	1.26	2.88 ± 3.4	2.06	1.12 ± 0.5	0.97
	$I_f \oplus I_m$		3.06 ± 0.6*	3.01	5.36 ± 4.4	3.71	14.57 ± 9.4*	11.58	1.46 ± 1.3	1.12
	$I_f \oplus I_m \oplus S_m$		1.26 ± 0.4	1.20	2.08 ± 2.2	1.27	2.79 ± 1.6	2.45	1.05 ± 0.4	0.97
Reg	$I_f \oplus I_m$		1.43 ± 0.8*	1.29	1.71 ± 1.4*	1.37	2.44 ± 1.1*	2.17	3.40 ± 2.3*	2.71
	$I_f \oplus I_m \oplus S_m$		1.91 ± 1.3	1.59	1.92 ± 1.5	1.44	2.58 ± 1.1	2.33	3.88 ± 2.5	3.16
JRS-reg	$I_f \oplus I_m$		1.16 ± 0.3	1.16	1.32 ± 0.6	**1.11**	2.08 ± 1.0	1.82	2.57 ± 2.0	2.04
	$I_f \oplus I_m \oplus S_m$		1.20 ± 0.4	1.13	1.35 ± 0.7	1.16	2.08 ± 1.0	1.82	2.63 ± 2.3	1.90
Cross-stitch	$I_f \,\|\|\, I_f \oplus I_m$	Segmentation	1.47 ± 0.3*	1.48	2.93 ± 3.0*	2.08	2.93 ± 2.0*	2.25	1.19 ± 1.0	0.89
		Registration	1.10 ± 0.3	1.07	1.38 ± 0.7	1.17	2.12 ± 1.0	1.89	2.55 ± 2.1	1.89
	$I_f \,\|\|\, I_f \oplus I_m \oplus S_m$	Segmentation	**1.06 ± 0.3**	**0.99**	**1.27 ± 0.4**	1.15	**1.76 ± 0.8**	**1.47**	**0.91 ± 0.4**	**0.82**
		Registration	1.10 ± 0.3	1.06	1.30 ± 0.6	1.13	2.00 ± 1.0	1.75	2.45 ± 2.1	1.81
	$I_f \oplus S_m \,\|\|\, I_f \oplus I_m \oplus S_m$	Segmentation	2.05 ± 0.7*	2.00	3.66 ± 4.4*	2.19	2.44 ± 1.0*	2.35	1.09 ± 0.5*	0.93
		Registration	1.40 ± 0.4	1.35	1.31 ± 0.6	1.17	2.27 ± 1.0	2.02	2.56 ± 1.9	1.96
	$I_f \oplus I_m \oplus S_m \,\|\|\, I_f \oplus I_m \oplus S_m$	Segmentation	1.08 ± 0.3	1.05	1.54 ± 0.9*	1.28	1.88 ± 1.0	1.61	1.01 ± 0.7	0.82
		Registration	1.20 ± 0.3	1.18	1.35 ± 0.7	1.16	2.12 ± 1.1	1.87	2.54 ± 2.2	1.80

(MSD) and Hausdorff Distance (HD) defined as follows:

$$\mathrm{MSD} = \frac{1}{2}\left(\frac{1}{N}\sum_{i=1}^{n} d\left(a_i, S_g\right) + \frac{1}{M}\sum_{i=1}^{m} d\left(b_i, S_f\right)\right), \tag{5.10}$$

$$\mathrm{HD} = \max\left\{\max_i\left\{d\left(a_i, S_g\right)\right\}, \max_j\left\{d\left(b_i, S_f\right)\right\}\right\}, \tag{5.11}$$

where $\{a_1;\ a_2;\ \ldots;\ a_n\}$ and $\{b_1;\ b_2;\ \ldots;\ b_m\}$ are the surface mesh points of the manual and generated contours, respectively, and $d\left(a_i, S_g\right) = \min_j \|b_j - a_i\|$. For all the experiments, we apply the largest connected component operation on the network prediction.

In order to evaluate the quality of the deformations, we calculate the determinant of the Jacobian matrix. A Jacobian of 1 indicates that no volume change has occurred; a Jacobian > 1 indicates expansion, a Jacobian between 0 and 1 indicates shrinkage, and a Jacobian ≤ 0 indicates a singularity, i.e. a place where folding has occurred. We can quantify the smoothness and quality of the DVF by indicating the fraction of foldings per image and by calculating the standard deviation of the Jacobian alongside the MSD of the segmentation.

A repeated one-way ANOVA test was performed using a significance level of $p = 0.05$. P-values are only stated for the comparisons between the best network with the other networks.

5.4 Experiments and results

In the paper we present two single-task networks dubbed *Seg* and *Reg* networks (see Sections 5.2.2.1 and 5.2.2.2 for more details). Moreover, we investigated multiple multi-task networks, namely JRS-reg, dense, SEDD, and Cross-stitch (see Sections 5.2.3.1, 5.2.3.2, 5.2.3.3, and 5.2.3.4 for more details). We compared our proposed methods against three state-of-the-art methods that were developed for prostate CT contouring. These methods represent three approaches, namely an iterative conventional registration method, a deep learning-based registration method, and a hybrid method. For the iterative method, we used `elastix` software [128] with the NCC similarity loss using the settings proposed by Qiao *et. al.* [131]. In the deep learning method proposed by Elmahdy *et. al.* [17], a generative network is trained for contour propagation by registration, while a discrimination network evaluates the quality of the propagated contours. Finally, we compare our methods against the hybrid method proposed by Elmahdy *et. al.* [23], where a CNN network segments the bladder and then feeds it to the iterative registration method as prior knowledge.

Following, we optimize some of the network settings on the validation set (HMC), in order to investigate the influence of the bending energy weight, network inputs, weighting strategy and network architecture on the results. Then, on the independent test set, we present the final results comparing with methods from the literature.

5.4.1 Bending energy weight

We compared the single-task registration, the JRS-reg method and the Cross-stitch network for a set of bending energy weights, see Equations (5.4) and (5.6), while the weights of the other loss functions are set to 1. Figure 5.2 shows the performance of the aforementioned methods using different bending energy weights. The optimal performance of the registration network occurs at a bending weight of 0.5, while the optimal bending weight for both JRS-reg and Cross-stitch network is much lower but with higher standard deviation of the Jacobian. Therefore, for the remainder of the paper we set the weight of the bending energy to 0.5 since it achieves the best compromise between the contour performance in terms of MSD and the registration performance in terms of the std. of the Jacobian determinant.

5.4.2 Optimization of the networks inputs

During training, validation, and testing, we have access to the fixed image I_f, the moving image I_m, and the moving segmentation S_m. In Table 5.1 we compared different sets of inputs on the validation dataset. This experiment helps to better understand how these network interpret and utilize these inputs and how this would reflect on the network outcome represented by the MSD metric. For this experiment we used equal loss weights for the MTL networks.

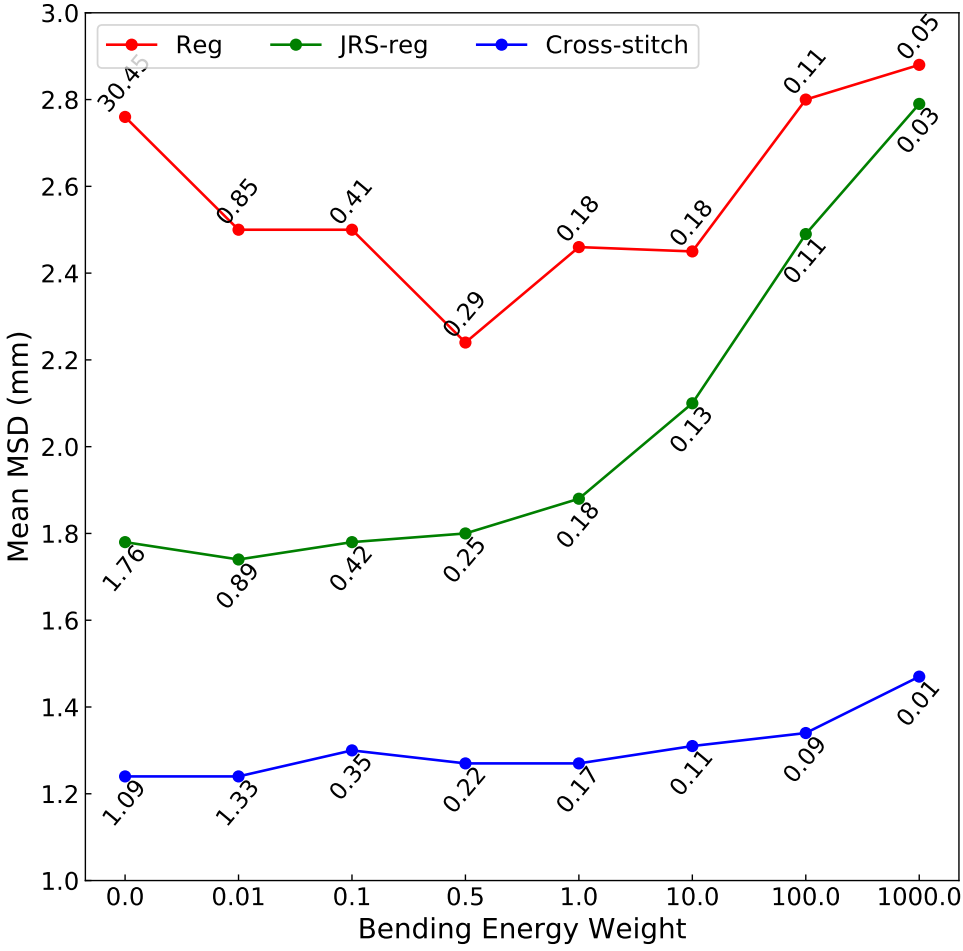

Figure 5.2: The performance of the registration, JRS-registration and Cross-stitch networks with different bending energy weights on the validation set (HMC), in terms of mean MSD averaged over the four organs. The annotation at each point represents the standard deviation of the determinant of the Jacobian.

Feeding S_m to the segmentation network improves the results substantially compared to only feeding I_f, especially for the seminal vesicles, while feeding I_m deteriorates the results. For the registration and JRS-reg networks, feeding S_m alongside I_f and I_m resulted in a similar performance compared to not feeding it. Since the Cross-stitch network is composed of two networks, one for segmentation and the other for registration, we experimented with various combinations of inputs. The results are very consistent with our previous findings on the single-task networks on the effect of

Table 5.2: MSD (mm) values for the different networks and loss weighting methods for the HMC dataset. Lower values are better. Stars and daggers denote one-way ANOVA statistical significance for inter-network experiments with respect to Homoscedastic weights and intra-network experiments with respect to Cross-stitch with Equal weights, respectively. Grey numbers represent the values of the worst path between the segmentation and registration paths, while bold numbers represent the best results.

			Prostate		Seminal vesicles		Rectum		Bladder	
Network	Weight	Output path	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median
JRS-reg	Equal	Registration	1.20 ± 0.4	1.13	1.35 ± 0.7	1.16	2.08 ± 1.0	1.82	$2.63 \pm 2.3^{*}$	1.90
	Homoscedastic	Registration	1.20 ± 0.3	1.20	1.22 ± 0.5	1.07	2.05 ± 1.0	1.81	2.34 ± 2.2	1.60
	DWA	Registration	1.22 ± 0.3	1.18	$1.37 \pm 0.7^{*}$	1.20	$2.29 \pm 1.1^{*}$	2.04	$3.18 \pm 2.4^{*}$	2.43
Dense	Equal	Segmentation	1.14 ± 0.4	1.06	1.73 ± 2.1	1.12	1.91 ± 0.9	1.64	1.04 ± 0.7	0.87
		Registration	1.20 ± 0.3	1.11	$1.33 \pm 0.7^{*}$	1.10	2.16 ± 1.1	1.85	2.56 ± 1.9	1.90
	Homoscedastic	Segmentation	1.09 ± 0.3	1.04	1.51 ± 1.2	1.13	1.86 ± 0.8	1.69	0.99 ± 0.4	0.91
		Registration	1.17 ± 0.3	1.15	1.31 ± 0.6	1.13	2.17 ± 1.0	1.96	$2.63 \pm 2.0^{*}$	1.95
	DWA	Segmentation	$1.12 \pm 0.3^{*\dagger}$	1.04	1.74 ± 2.0	1.13	$1.99 \pm 0.9^{*}$	1.77	1.00 ± 0.4	0.85
		Registration	1.14 ± 0.3	1.14	1.27 ± 0.6	**1.07**	$2.24 \pm 1.1^{*}$	1.97	2.72 ± 1.9	2.13
SEDD	Equal	Segmentation	$1.47 \pm 0.6^{*\dagger}$	1.31	2.81 ± 4.6	1.34	1.97 ± 1.0	1.59	1.21 ± 1.0	0.94
		Registration	$1.28 \pm 0.4^{*}$	1.19	$1.50 \pm 0.9^{*}$	1.26	$2.26 \pm 1.1^{*}$	1.94	$2.61 \pm 2.1^{*}$	1.83
	Homoscedastic	Segmentation	$1.15 \pm 0.3^{\dagger}$	1.14	1.47 ± 1.0	1.22	2.12 ± 1.1	1.91	0.99 ± 0.2	0.94
		Registration	1.19 ± 0.3	1.21	1.23 ± 0.5	1.13	2.15 ± 1.0	1.92	2.31 ± 2.0	1.64
	DWA	Segmentation	$1.22 \pm 0.3^{*\dagger}$	1.18	1.44 ± 0.8	1.21	2.12 ± 1.4	1.73	1.10 ± 0.6	0.93
		Registration	1.22 ± 0.3	1.22	$1.32 \pm 0.6^{*}$	1.10	$2.30 \pm 1.1^{*}$	2.01	$2.86 \pm 1.9^{*}$	2.41
Cross-stitch	Equal	Segmentation	$\mathbf{1.06 \pm 0.3}$	**0.99**	$\mathbf{1.27 \pm 0.4}$	1.15	$\mathbf{1.76 \pm 0.8}$	**1.47**	$\mathbf{0.91 \pm 0.4}$	**0.82**
		Registration	$1.10 \pm 0.3^{*}$	1.06	1.30 ± 0.6	1.13	$2.00 \pm 1.0^{*}$	1.75	2.45 ± 2.1	1.81
	Homoscedastic	Segmentation	$1.23 \pm 0.3^{\dagger}$	1.16	1.51 ± 1.2	1.17	2.37 ± 1.0	2.09	0.92 ± 0.2	0.89
		Registration	1.24 ± 0.3	1.24	1.32 ± 0.6	1.13	2.12 ± 1.0	1.89	2.45 ± 1.9	1.97
	DWA	Segmentation	$1.34 \pm 0.4^{*\dagger}$	1.27	1.75 ± 1.7	1.29	$2.32 \pm 0.9^{\dagger}$	2.11	$1.17 \pm 0.8^{*}$	0.91
		Registration	1.22 ± 0.3	1.19	1.27 ± 0.6	1.09	$2.21 \pm 1.0^{*}$	2.00	$2.93 \pm 2.3^{*}$	2.27

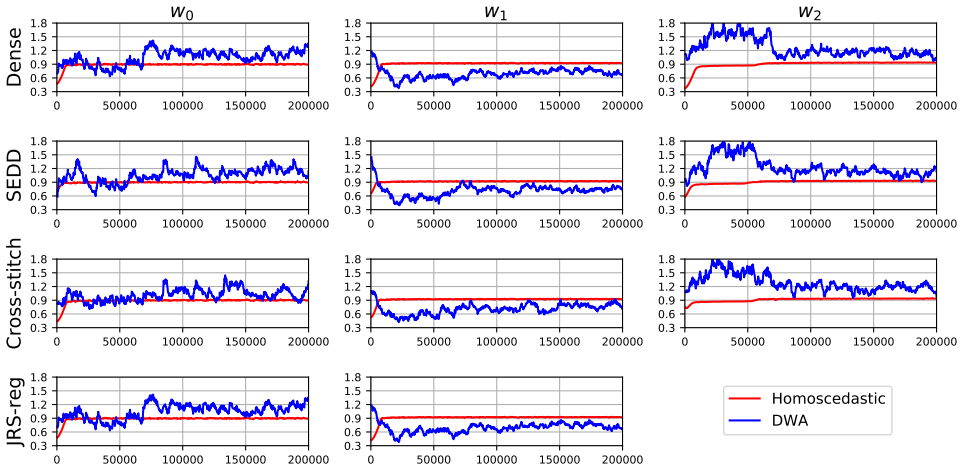

Figure 5.3: The evolution of the loss weights during training for different multi-task networks on the validation dataset (HMC).

using S_m as an input.

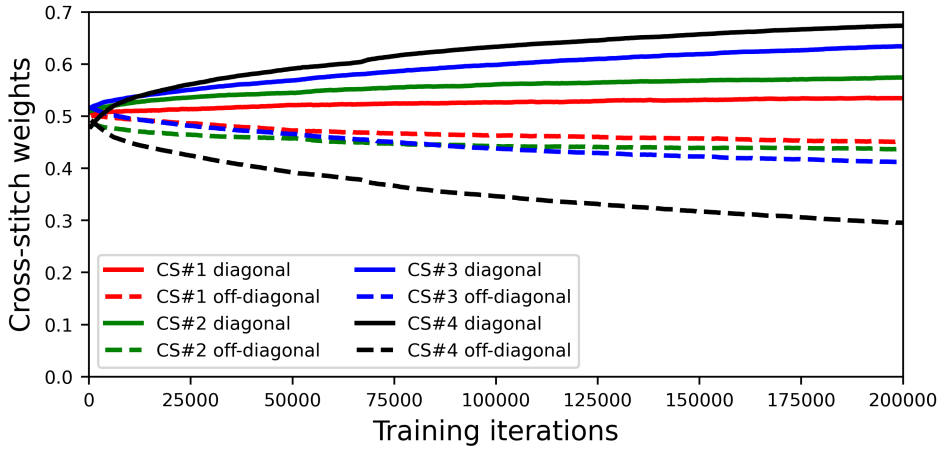

Figure 5.4: The evolution of the Cross-stitch units weights during training using equal weights. CS#1 and CS#2 are placed in the down-sampling path, while CS#3 and CS#4 are placed in the upsampling path. The solid lines represent the mean of the weights across the diagonal of the CS unit, while the dashed lines represent the mean of the off-diagonal weights.

For the remainder of this paper, we chose to use I_f as input for the segmentation network, and I_f and I_m as inputs for the registration network. Although adding S_m proved to be better especially for the segmentation network, here we exclude it, since these two methods act as a baseline and this is the standard setting in single-task networks. For dense, SEDD, and JRS-reg networks, we select a concatenation of I_m, I_f, and S_m for the final network. For the Cross-stitch network, we select I_f for the segmentation network and the concatenation of I_m, I_f, and S_m for the registration network.

5.4.3 Optimization of loss weighting strategy

In this experiment we investigate the performance of the various loss weighting strategies introduced in Section 5.2.4 in order to select the best weighting method for the underlying tasks.

Table 5.2 shows the results of the different weighting strategies for the MTL networks in terms of MSD. For the JRS-reg network architecture, weighting the losses with homoscedastic uncertainty achieved comparable results to using equal weights, while DWA scored somewhat less. For the dense and SEDD architectures, homoscedastic weighting achieved a slightly better performance, while equal weights was best for the Cross-stitch network. For these architectures (dense, SEDD, and

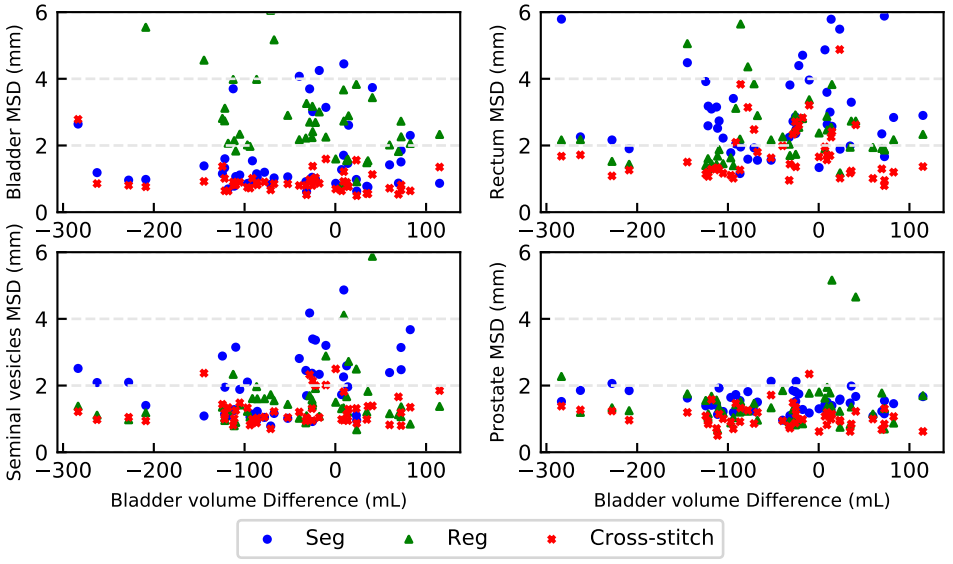

Figure 5.5: The effect of the bladder volume deviation from the planning volume on the performance of the Seg, Reg, and Cross-stitch networks for the validation set (HMC).

Cross-stitch), the segmentation output path showed improvement over the registration output path.

Figure 5.3 illustrates the evolution of the loss weights w_i during training, for different multi-task network architectures and weighting strategies.

For the remainder of this paper and based on the previous findings, we chose the homoscedastic uncertainty weighting strategy for the JRS-reg, dense and SEDD networks, while using equal weights for the Cross-stitch network.

5.4.4 Analysis of cross-stitch units

Analysis of the behavior of the Cross-stitch units during training facilitates the understanding of how the segmentation and registration networks interacts in the MTL settings. Figure 5.4 shows the mean of the CS units across the diagonal and off-diagonal (See Equation (5.5)). Higher weights on the diagonal means that the network tends to separate the task-specific feature maps, while higher weights off-diagonal means that the network tends to share the corresponding feature maps.

5.4.5 Effect of the bladder filling

For the HMC dataset, which was used for training and validation, a bladder filling protocol was in place, meaning that the deformation of the bladder between daily and

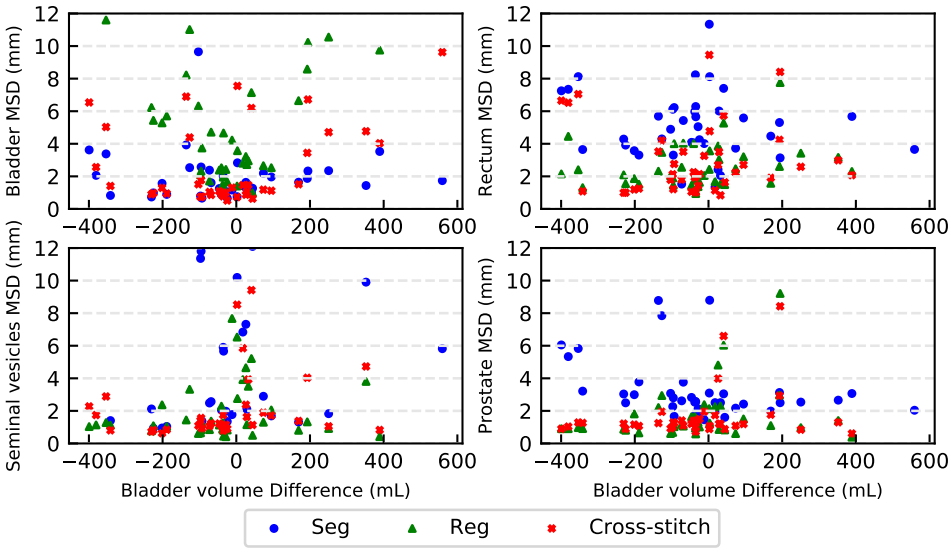

Figure 5.6: The effect of the bladder volume deviation from the planning volume on the performance of the STL and the Seg, Reg, and Cross-stitch networks for the independent test set (EMC).

planning scans is not large. However, this is not the scenario for the EMC dataset, the test set.

Figure 5.5 and 5.6 illustrates the effect of the bladder volume variation from the planning scan on the performance of the Seg, Reg, and Cross-stitch networks. The Cross-stitch network is resilient to bladder filling for both the HMC and EMC datasets.

5.4.6 Evaluation of the quality of the DVF

The smoothness of the predicted DVF is an important parameter to evaluate the predicted deformation field. Table 5.5 shows a detailed analysis of the DVF in terms of the standard deviation of the determinant of the Jacobian as well as the folding fraction for the registration path of the different networks.

5.4.7 Comparison against the state-of-the-art

Table 5.3 and 5.4 show the results for the validation set (HMC) and test set (EMC), respectively. The first two networks in each table are single-task networks. For both sets, the registration network outperformed the segmentation network for all organs except the bladder. The mean MSD for the independent test set is higher than the corresponding numbers in the validation set for most organs. However, the median values are on par. For the MTL networks, the segmentation path of the networks

Table 5.3: MSD (mm) values for the different networks on the validation set (HMC). Lower values are better.

		Prostate		Seminal vesicles		Rectum		Bladder	
Network	Output path	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median
Seg	Segmentation	1.49 ± 0.3	1.49	2.50 ± 2.6	2.09	3.39 ± 2.2	2.73	1.60 ± 1.1	1.13
Reg	Registration	1.43 ± 0.8	1.29	1.71 ± 1.4	1.37	2.44 ± 1.1	2.17	3.40 ± 2.3	2.71
JRS-reg	Registration	1.20 ± 0.3	1.20	1.22 ± 0.5	**1.07**	2.05 ± 1.0	1.81	2.34 ± 2.2	1.60
Dense	Segmentation	1.09 ± 0.3	1.04	1.51 ± 1.2	1.13	1.86 ± 0.8	1.69	0.99 ± 0.4	0.91
	Registration	1.17 ± 0.3	1.15	1.31 ± 0.6	1.13	2.17 ± 1.0	1.96	2.63 ± 2.0	1.95
SEDD	Segmentation	1.15 ± 0.3	1.14	1.47 ± 1.0	1.22	2.12 ± 1.1	1.91	0.99 ± 0.2	0.94
	Registration	1.19 ± 0.3	1.21	1.23 ± 0.5	1.13	2.15 ± 1.0	1.92	2.31 ± 2.0	1.64
Cross-stitch	Segmentation	$\mathbf{1.06 \pm 0.3}$	**0.99**	$\mathbf{1.27 \pm 0.4}$	1.15	$\mathbf{1.76 \pm 0.8}$	**1.47**	$\mathbf{0.91 \pm 0.4}$	**0.82**
	Registration	1.10 ± 0.3	1.06	1.30 ± 0.6	1.13	2.00 ± 1.0	1.75	2.45 ± 2.1	1.81
Elastix [131]	Registration	1.73 ± 0.7	1.59	2.71 ± 1.6	2.45	3.69 ± 1.2	3.50	5.26 ± 2.6	4.72
Hybrid [23]	Registration	1.27 ± 0.3	1.25	1.47 ± 0.5	1.32	2.03 ± 0.6	1.85	1.75 ± 1.0	1.26
JRS-GAN [17]	Registration	1.14 ± 0.3	1.04	1.75 ± 1.3	1.44	2.17 ± 1.1	1.89	2.25 ± 1.9	1.54

Table 5.4: MSD (mm) values for the different networks on the independent test set (EMC). Lower values are better. Results for JRS-GAN are not available for this dataset.

		Prostate		Seminal vesicles		Rectum		Bladder	
Network	Output path	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median	$\mu \pm \sigma$	median
Seg	Segmentation	3.18 ± 1.8	2.57	9.33 ± 10.1	5.82	5.79 ± 3.4	5.18	1.88 ± 1.5	1.50
Reg	Registration	2.01 ± 2.5	1.18	2.86 ± 5.2	1.18	2.89 ± 2.5	2.23	5.98 ± 4.7	4.44
JRS-reg	Registration	1.94 ± 2.6	1.16	2.48 ± 4.8	1.01	2.67 ± 2.4	2.05	4.80 ± 4.6	2.12
Dense	Segmentation	2.01 ± 2.6	**1.15**	4.08 ± 7.2	1.23	3.70 ± 5.4	**2.03**	2.75 ± 3.1	1.23
	Registration	1.93 ± 2.5	1.15	2.53 ± 4.7	1.01	2.67 ± 2.3	2.13	5.08 ± 4.4	3.01
SEDD	Segmentation	1.99 ± 2.4	1.24	6.26 ± 8.9	3.01	4.21 ± 4.9	2.12	2.43 ± 2.9	1.04
	Registration	1.92 ± 2.5	1.19	2.43 ± 4.5	1.07	2.72 ± 2.4	2.17	4.86 ± 4.4	2.22
Cross-stitch	Segmentation	1.88 ± 1.9	1.30	2.76 ± 3.5	1.28	4.87 ± 6.8	2.49	$\mathbf{1.66 \pm 1.7}$	**0.85**
	Registration	1.91 ± 2.3	1.23	2.41 ± 4.5	**0.95**	2.78 ± 2.4	2.16	4.90 ± 4.0	2.84
Elastix [131]	Registration	$\mathbf{1.42 \pm 0.7}$	1.17	2.07 ± 2.6	1.24	3.20 ± 1.6	3.07	5.30 ± 5.1	3.27
Hybrid [23]	Registration	1.55 ± 0.6	1.36	$\mathbf{1.65 \pm 1.3}$	1.22	$\mathbf{2.65 \pm 1.6}$	2.36	3.81 ± 3.6	2.26

Table 5.5: Analysis of the determinant of the Jacobian for the validation and the independent test sets. Lower values are better.

	Validation set (HMC)		Independent test set (EMC)	
Network	Std. Jacobian	Folding fraction	Std. Jacobian	Folding fraction
Reg	0.2935 ± 0.1022	0.0049 ± 0.0039	0.4129 ± 0.2258	0.0112 ± 0.0115
JRS-reg	0.2543 ± 0.0505	0.0030 ± 0.0014	0.3148 ± 0.1106	0.0066 ± 0.0062
Dense	0.2062 ± 0.0431	0.0018 ± 0.0012	0.2558 ± 0.0899	0.0036 ± 0.0027
SEDD	0.2626 ± 0.1167	0.0019 ± 0.0016	0.4287 ± 0.3000	0.0066 ± 0.0074
Cross-stitch	0.2241 ± 0.0784	0.0024 ± 0.0018	0.3301 ± 0.1869	0.0071 ± 0.0070

achieved better performance than the registration path on both datasets except for the seminal vesicles. The Cross-stitch network achieved the best results compared to the other MTL networks.

The proposed STL and MTL networks were compared against other state-of-the-art methods that were evaluated using the HMC dataset. For the validation set, the STL network achieved comparable results, while the Cross-stitch network outperformed

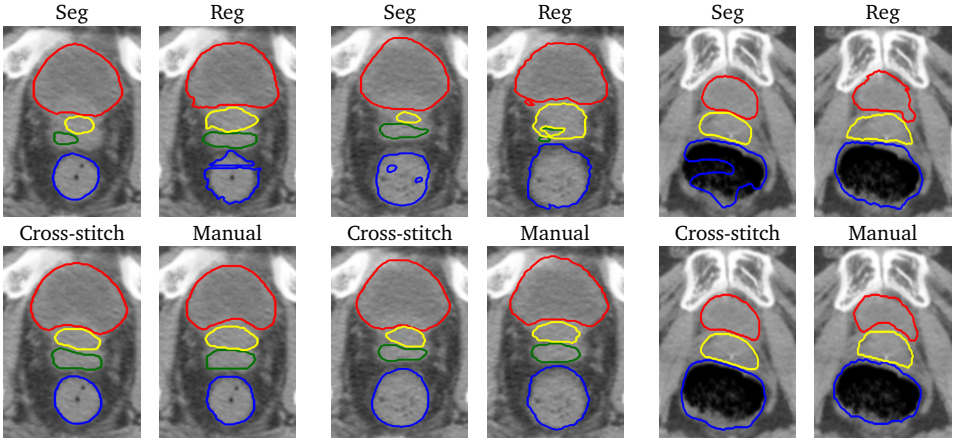

Figure 5.7: Example contours from the validation dataset (HMC) generated by the proposed STL and MTL networks. From left to right, the selected cases are the first, second, and third quartile in terms of the prostate MSD of the Cross-stitch network. The contours of the bladder, prostate, seminal vesicles, and rectum are colored in red, yellow, green, and blue, respectively.

these methods for both output paths. On the test set, *elastix* [131] and the Hybrid method [23] performed better except for the bladder, although the median values of the MTL networks were better.

For the quality of the predicted contours, Figure 5.7 and 5.8 show example contours from the HMC and EMC datasets for the Seg, Reg, and Cross-stitch networks. The examples show that the Cross-stitch network achieves better results compared to the Seg and Reg networks especially for the seminal vesicles and rectum with large gas pockets.

5.5 Discussion

In this study, we proposed to merge image registration and segmentation on the architectural level as well as the loss, via a multi-task learning setting in order to leverage their strengths and mitigate their weaknesses through the sharing of beneficial information. We studied different network architectures and loss weighting methods in order to explore how these tasks interact, and thereby leverage the shared knowledge between them. Moreover, we carried out extensive quantitative analysis in the context of adaptive radiotherapy, and compared the proposed multi-task methods to their single-task counterparts. In this paper, a substantial number of experiments were executed, where we explored the following methodological choices: the bending energy weight, the input to the STL and MTL networks, and the loss weighting method. We also performed a thorough analysis on how Cross-stitch units and loss

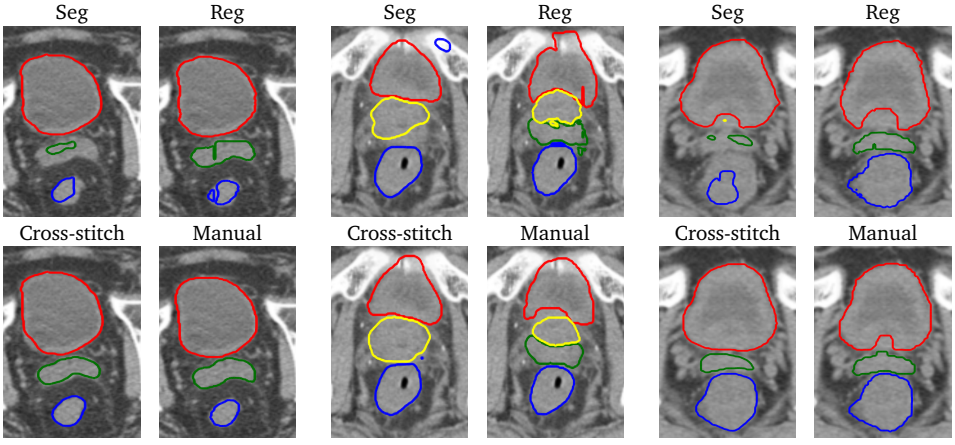

Figure 5.8: Example contours from the independent test set (EMC) generated by the proposed STL and MTL networks. From left to right, the selected cases are the first, second, and third quartile in terms of the prostate MSD of the Cross-stitch network.

weights evolve during training. Finally, we compared our proposed methods against state-of-the-art methods.

In all the experiments we fixed the weight of the bending energy weight so that the network would not set it too low in order to improve the DSC of the deformed contours on the account of the smoothness of the predicted DVF. As shown in Figure 5.2 low bending energy weights result in better contour quality on the account of the smoothness of the predicted DVF.

For the inputs to the STL networks, additionally feeding S_m to the segmentation network resulted in a statistically significant improvement especially for the seminal vesicles. Apparently the network considers S_m as an initial estimation for S_f and subsequently uses it as a guidance for its final prediction. When feeding $I_m$the results deteriorated; this may confuse the network as I_f and I_m have the same anatomy but with different shapes and local positions. The addition of both I_m and S_m performed similar to the addition of only S_m, which indicates that the networks learned to ignore I_m. For the registration network, the addition of S_m resulted in a sub-optimal result, since the S_m contours on its own does not represent the underlying deformation well.

For the inputs to the MTL networks, in the JRS-reg network, feeding S_m alongside I_f and I_m resulted in a similar performance compared to not feeding it. This indicates that the incorporation of S_m via the DSC loss, already enables the JRS-reg network to exploit this extra information, and that additionally adding S_m as a network input does not provide further benefits. In the Cross-stitch network, we found that adding S_m to the registration network results in a statistically significant improvement. Furthermore, feeding S_m to one of the networks is sufficient, proving that segmentation

and registration networks communicate their knowledge efficiently through the Cross-stitch units.

We selected the STL networks with I_f (for segmentation) and I_f alongside I_m (for registration) as input to our baseline methods. Between these two networks, the registration network performed better overall, since the registration network leverages prior knowledge from the organs in the moving image. For the bladder, the segmentation network achieved better results; Apparently the registration network had difficulties finding the correspondence between the bladder in the fixed and moving images, since it tends to deform considerably between visits. However, the segmentation network failed to segment the seminal vesicles for five cases. That is explained by the fact that the seminal vesicles is a difficult structure to segment, due to its relatively small size, undefined borders, and poor contrast with its surroundings. The registration network on the other hand is able to employ the surrounding anatomy as context, to accurately warp the seminal vesicles.

For the multi-task networks, we demonstrated that fusing segmentation and registration tasks is performing better than its single-task counterparts. Merging these tasks using Cross-stitch network achieved the best results on both the validation and testing datasets.

Different loss weighting methods achieved comparable results as shown in Table 5.2. In Figure 5.3, homoscedastic uncertainty tended to weigh all losses equally, using almost a fixed weight of 0.9 during most of the training iterations. On the contrary, DWA tended to fluctuate during training as the weights are updated based on the ratio of the loss from previous iterations, which fluctuates due to the batch-based training. Since the fixed and moving images are affinely registered beforehand, DWA tended to down-weigh the registration loss and the associated DSC at the beginning of the training, while weighting the segmentation network loss more in order to improve its prediction. Later during training, all the weights stabilized around 0.9 similar to homoscedastic uncertainty. Although both methods stabilized by the end of the training around the same value (0.9), the homoscedastic uncertainty achieved slightly better results compared to DWA and equal weighting methods, except for the Cross-stitch network. Our reasoning behind this is that homoscedastic uncertainty, unlike other methods, is learnable during the training and highly dependent on the underlying task uncertainty.

By analyzing the performance of the Cross-stitch units as demonstrated in Figure 5.4, we found that the Cross-stitch units tended to average feature maps for the down-sampling path, while preferring to be more task-specific for the upsampling path. This somewhat mimics the shared encoder double decoder (SEDD) network, but in contrast to this network, the Cross-stitch network does not completely split the decoder paths. This finding confirms that the segmentation and registration tasks are

correlated and thereby encode similar features.

We carried out an experiment to study the effect of the bladder filling protocol between the HMC and EMC datasets. As shown in Figure 5.5, the HMC dataset has a bladder filling protocol so the volume of the bladder changes slightly around 100 mL between different sessions, which is not the case for the EMC dataset as shown in Figure 5.6. Since the registration-based networks and joint networks were trained on small bladder deformations, they failed on large deformations, however the segmentation network was not affected since it does not depend on the deformation but rather the underlying texture to segment the bladder.

In terms of the smoothness of the predicted DVF shown in Table 5.5, MTL networks achieved lower numbers for the standard deviation of the Jacobian as well as for the folding fraction, compared to the STL network (Reg), on both the test and validation set. Our reasoning is that joining the segmentation task to the registration task works as an additional regularization to the registration network. Due to the fact that the higher the quality of the predicted DVF, the higher the quality of the propagated contours and subsequently the lower the DSC loss. The numbers on the test set are slightly higher than the validation set, but this is due to the variance between the deformations between both sets and the fact that the network has not seen the test set before. This can be addressed using transfer learning as suggested by Elmahdy *et al.* [106] or by using synthetic deformations that mimic the one presented in the EMC dataset.

In the paper, we compared our algorithm against different algorithms from various categories: non-learning (`elastix` [128], a popular conventional tool); hybrid [23], and GAN-based [17]. The presented multi-task networks outperformed these approaches on the validation set and performed on par to these methods for the test set. However, the test time for the hybrid and `elastix` methods are in the order of minutes, while the presented methods have the advantage of fast prediction in less than a second. This enables online automatic re-contouring of daily scans for adaptive radiotherapy. Moreover, in our hybrid study [23] we carried out an extensive dosimetric evaluation alongside the geometric evaluation. The predicted contours from that study met the dose coverage constraints in 86%, 91%, and 99% of the cases for the prostate, seminal vesicles, and lymph nodes, respectively. Since our multi-task networks outperformed the geometrical results in that study, we expect that our contours would achieve a higher success rate in terms of the dose coverage. This could potentially reduce treatment related complications and therefore improve patient quality-of-life after treatment.

A promising direction for future research is the addition of a third task, potentially radiotherapy dose plan estimation. Hence, we can generate contours that are consistent with an optimal dose planning. Further studies could also focus on sophisticated

MTL network architectures similar to sluice networks [132] or routing networks [133]. Moreover, we can study how to fuse the contours from the segmentation and registration paths in a smarter way rather than simply selecting one of them based on the validation set.

5.6 Conclusion

In this paper, we propose to formulate the registration and segmentation tasks as a multi-task learning problem. We presented various approaches in order to do so, both on an architectural level and via the loss function. We experimented with different network architectures in order to investigate the best setting that maximizes the information flow between these tasks. Moreover, we compared different loss weighting methods in order to optimally combine the losses from these tasks.

We proved that multi-task learning approaches outperform their single-task counterparts. Using an adaptive parameter sharing mechanism via Cross-stitch units gives the networks freedom to share information between these two tasks, which resulted in the best performance. An equal loss weighting approach had similar performance to more sophisticated methods.

The cross stitch network with equal loss weights achieved a median MSD of 0.99 mm, 0.82 mm, 1.13 mm and 1.47 mm on the validation set and 1.09 mm, 1.24 mm, 1.02 mm, and 2.10 mm on the independent test set for the prostate, bladder, seminal vesicles, and rectum, respectively. That is equal or less than slice thickness (2 mm). Due to the fast inference of the methods, the proposed method is highly promising for automatic re-contouring of follow-up scans for adaptive radiotherapy, potentially reducing treatment related complications and therefore improving patient quality-of-life after treatment.

5.7 Acknowledgment

The HMC dataset with contours was collected at Haukeland University Hospital, Bergen, Norway, and was provided to us by responsible oncologist Svein Inge Helle and physicist Liv Bolstad Hysing. The EMC dataset with contours was collected at Erasmus University Medical Center, Rotterdam, The Netherlands, and was provided to us by radiation therapist Luca Incrocci and physicist Mischa Hoogeman. They are gratefully acknowledged.