Medical Image Analysis 95 (2024) 103181

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Active learning using adaptable task-based prioritisation

Shaheer U. Saeed [a,*], João Ramalhinho [a], Mark Pinnock [a], Ziyi Shen [a], Yunguan Fu [a,b], Nina Montaña-Brown [a], Ester Bonmati [a,c], Dean C. Barratt [a], Stephen P. Pereira [a,d], Brian Davidson [a,e], Matthew J. Clarkson [a], Yipeng Hu [a]

[a] Centre for Medical Image Computing, Wellcome/EPSRC Centre for Interventional & Surgical Sciences, and Department of Medical Physics & Biomedical Engineering, University College London, London, UK
[b] InstaDeep, London, UK
[c] School of Computer Science and Engineering, University of Westminster, London, UK
[d] Institute for Liver and Digestive Health, University College London, London, UK
[e] Centre for Surgical Innovation, Organ Regeneration and Transplantation (CISORT), Division of Surgery & Interventional Science, University College London, London, UK

ARTICLE INFO

MSC:
41A05
41A10
65D05
65D17

Keywords:
Active Learning
Medical Image Quality
Segmentation

ABSTRACT

Supervised machine learning-based medical image computing applications necessitate expert label curation, while unlabelled image data might be relatively abundant. Active learning methods aim to prioritise a subset of available image data for expert annotation, for label-efficient model training. We develop a controller neural network that measures priority of images in a sequence of batches, as in batch-mode active learning, for multi-class segmentation tasks. The controller is optimised by rewarding positive task-specific performance gain, within a Markov decision process (MDP) environment that also optimises the task predictor. In this work, the task predictor is a segmentation network. A meta-reinforcement learning algorithm is proposed with multiple MDPs, such that the pre-trained controller can be adapted to a new MDP that contains data from different institutes and/or requires segmentation of different organs or structures within the abdomen. We present experimental results using multiple CT datasets from more than one thousand patients, with segmentation tasks of nine different abdominal organs, to demonstrate the efficacy of the learnt prioritisation controller function and its cross-institute and cross-organ adaptability. We show that the proposed adaptable prioritisation metric yields converging segmentation accuracy for a new kidney segmentation task, unseen in training, using between approximately 40% to 60% of labels otherwise required with other heuristic or random prioritisation metrics. For clinical datasets of limited size, the proposed adaptable prioritisation offers a performance improvement of 22.6% and 10.2% in Dice score, for tasks of kidney and liver vessel segmentation, respectively, compared to random prioritisation and alternative active sampling strategies.

1. Introduction

Medical imaging tasks are increasingly being automated using machine learning by utilising expert annotated data (Lee et al., 2017; Erickson et al., 2017). Supervised learning using expert annotations allows for reliable predictions from the trained model, however, this expert annotation may often be expensive. Applications such as complex surgical planning thus become challenging to develop, due to the need for many structures to be annotated at the voxel-level and different regions of interest (ROIs) required by subsequent procedures, mandated by local expertise and protocols. This is further complicated by the now well-known problem of generalisation from deep models across different institutions, all of which are often under data size constraints.

Active learning (AL) aims to directly address the expensive data labelling by prioritising a subset of available unlabelled data for annotation, such that the machine learning models trained with these annotated data reach a predefined, or the same, performance level with fewer labelled samples, as models trained with all data being labelled. The efficiency of the performance convergence measures performance of the AL methods, in terms of the quantity of annotated data, i.e. required number of *AL iterations*, often compared with random sampling without prioritisation (Budd et al., 2021; Settles, 2009). Therefore, metrics that valuate how data samples affect AL convergence (hereinafter referred to as prioritisation metrics) are the key to the goal of fast convergence, i.e. using as few labelled samples as possible. Informativeness

* Corresponding author.
E-mail address: shaheer.saeed.17@ucl.ac.uk (S.U. Saeed).

https://doi.org/10.1016/j.media.2024.103181
Received 20 January 2023; Received in revised form 3 April 2024; Accepted 12 April 2024
Available online 16 April 2024

and representativeness are regarded as the main criteria in existing prioritisation metrics (Budd et al., 2021).

Informativeness estimates information gained if a particular labelled sample is added to training. Uncertainty with respect to the given samples is often used to quantify the informativeness, as it measures the amount of uncertain, therefore likely unknown, information that could be learnt by including the samples. For tasks like image segmentation or object boundary delineation, a summation of the lowest class probabilities over all pixels can be used (Shannon, 1948; Lewis and Gale, 1994), while high class probabilities are assumed high prediction confidence. An ensemble of multiple models was proposed for quantifying uncertainty (Settles, 2009; Czolbe et al., 2021). Monte-Carlo Dropout-based uncertainty estimation (Gal and Ghahramani, 2016) was also proposed and may be viewed as a special case of ensemble methods.

Representativeness measures the similarity between data samples, such that an effective AL strategy can be designed for prioritising those samples that can efficiently represent many others (Budd et al., 2021; Zhou et al., 2017; Yoo and Kweon, 2019; Peng et al., 2022). Distances between multiple images have been proposed, for example, based on features extracted from a trained model for a different, usually unsupervised, task such as self-reconstruction (Yang et al., 2017; Smailagic et al., 2018; Ozdemir et al., 2018). Representativeness can also be combined with informativeness measures (Yang et al., 2017; Smailagic et al., 2018; Ozdemir et al., 2018).

However, general prioritisation metrics, such as Monte-Carlo Dropout and ensemble, have shown limited improvements in the number of iterations required for convergence compared to random sampling, especially for recent medical imaging datasets (Czolbe et al., 2021). Furthermore, we explore the efficacy of recent AL methods in Table 3. The fixed and non-adaptive nature of these metrics could lead to adverse consequences. For example, high uncertainty in samples may in fact be a result of label error or inconsistency, due to manual annotation difficulty (Czolbe et al., 2021). It has been speculated that not accounting for the impact of annotated samples, *post annotation*, and assuming that annotations are unambiguous and noise-free have led to the ineffective prioritisation metrics (Czolbe et al., 2021; Fang et al., 2017). This has been consistent with our preliminary results in a task of segmenting kidney on 3D CT images (summarised in Fig. 3(d), with further details discussed in Section 3). In contrast, task-based prioritisation can utilise task-specific feedback in formulating the prioritisation, such as the performance of a trained model for the subsequent task. This task-based feedback enables post annotation impact to be measured during model training (Fang et al., 2017; Woodward and Finn, 2017) and may alleviate the discussed limitations for individual tasks. As opposed to previous works that utilise prioritisation informed by task-specific measures e.g., through uncertainty metrics computed from deep learning task networks (Yang et al., 2017) or through learning a metric from the training set loss (Yoo and Kweon, 2019), our proposed framework learns prioritisation based on task-specific feedback directly. This is achieved using a reserved set of samples (see Section 2.4) for learning or adapting the AL prioritisation metric, during AL itself. The reserved set of samples, chosen randomly during training, is representative of the entire dataset. This formulation learns how performance is impacted given a particular selection of training data using reinforcement learning (RL), which utilises a reward based on performance impact of any given prioritisation, on this reserved reward computation set (referred to as the validation or support-validation set). This, task-performance-based prioritisation metric, learnt with the help of randomly selected samples allows us to capture the representativeness and the reward based on the task performance for this reserved set allows us to capture the informativeness.

In this work, we focus on organ segmentation on 3D abdominal CT images. Multi-organ segmentation or boundary delineation has a number of clinical applications (Fu et al., 2021; Gibson et al., 2018). Planning laparoscopic liver resection or liver surgery in general is one such example, in which localising the liver, liver vessels and surrounding anatomy is necessary for existing inter-modality image registration (Ramalhinho et al., 2022) and useful for subsequent navigation (Fusaglia et al., 2016) during the procedure. Moreover, AL will greatly benefit the development of automatic segmentation models for different clinical requirements, because of the potentially diverging protocol-specific needs, such as the types of vessels and/or organs required for different registration algorithms and changing local image-navigating procedures. We thus identify two aspects for a desirable AL approach in this application: (1) prioritising CT images to be annotated for the required ROI types (organs or anatomical structures), potentially new and unseen in developing such prioritisation strategy, and (2) the ability to adapt or generalise such prioritisation to image data from a different and novel institute.

We first propose a prioritisation metric based on direct feedback from the segmentation task using annotated samples, which is learnt using reinforcement learning (RL) based meta-learning. Second, we outline a mechanism, using the proposed meta-RL, to allow for the metric to be adapted to new data distributions including data from new institutes and for segmenting new ROI classes i.e. organs or structures unseen in training. In our formulation, task-based feedback for AL is delivered by means of a reward signal in the RL algorithm, in order to learn a prioritisation metric function. The reward signal is computed by measuring performance of a partially trained model on a set of samples for which annotations are available. The meta-RL further enables such prioritisation function to be useful across wider domains than with "simple" RL (Duan et al., 2016; Wang et al., 2017; Botvinick et al., 2019). In our framework, RL is used to first learn a task-based prioritisation scheme based on data selection, where data that leads to task performance gains is selected by the RL controller, for task-predictor training. This non-differentiable data selection together with demonstrated RL-based meta learning frameworks investigating similar non-differentiable problems are the reasons for using RL-based optimisation for our learnt AL metric. Meta-RL, with the use of a recurrent neural network (RNN) controller equips the system with adaptability over multiple downstream tasks, due to the ability of RNNs to enable learning and inferring using potentially useful sequential information, as demonstrated in Cotter and Conwell (1990), Santoro et al. (2016), Younger et al. (1999), Prokhorov et al. (2002), Hochreiter et al. (2001), Duan et al. (2016), Wang et al. (2017). Learning the adaptable AL metric is the first stage in our framework which involves learning optimal weights for the RNN controller using meta-RL. Once learnt, these optimal weights may be fixed and used for AL, where the controller adapts to any new task by utilising sequential information presented over multiple time-steps to the RNN, rather than relying on any weight updates, as explained further in Section 2.3.4.

It is important to highlight the difference between the proposed prioritisation metrics for AL and few-shot learning, which requires small number of annotated data from the novel classes and/or institutions during adaptation, e.g. Li et al. (2022), Feyjie et al. (2020). It is also interesting to compare our proposed methods with recent image quality assessment approaches. For example, although aiming for a distinct objective of prioritising data to label, the proposed prioritisation metrics share technical similarities with previous work that quantifies task amenability of samples using direct feedback from a clinical task such as organ segmentation (Saeed et al., 2021a, 2022a, 2021b, 2022b). Moreover, the proposed AL strategy is designed for medical images, as opposed to the language data used in previously proposed AL approaches that also utilised RL, Fang et al. (2017) and Woodward and Finn (2017). The algorithmic differences to these works (Pang et al., 2018; Fang et al., 2017; Woodward and Finn, 2017) are significant including problem definition, labelled example requirement, reward formulation and training methodology using meta-RL (rather than simple RL in previous works). Our work uses a reward based on a simultaneously trained task predictor, assumes a dynamically changing set for reward computation (offering potentially greater label efficiency when compared to methods that use large labelled sets for

rewards), can work for a number of machine learning problems such as segmentation, and allows for batch-mode active learning by modelling actions probabilistically, as opposed to the formulations presented in previous works which are often application-specific or designed for classification problems and may limit flexibility such as modelling actions as binary decisions rather than probabilistically and thus not allowing batch sampling or using large labelled sets for rewards (Pang et al., 2018; Fang et al., 2017; Woodward and Finn, 2017). We propose, for the first time, an adaptable AL training scheme where adaptability is equipped to the outer-level controller using meta-RL and additionally to the inner-level task predictor using Reptile, as outlined in Section 2.3.4. Using these adaptable functions we formulate an adaptable AL framework as outlined in Algo. 2, where we use part of the AL-prioritised samples for adaptation of the learnt AL and part for training the task predictor. Our proposed approach is an AL framework that aids training new task-predictors by prioritising samples to-be-labelled based on their predicted impact on the performed downstream task. This formulation may be used in practice by annotating only small subsets of high-priority samples, as determined by our proposed AL, as opposed to annotating entire datasets. This is different from few-shot meta-learning methods that focus on adapting the task-predictor to new datasets or tasks using randomly sampled data, but do not focus on determining sample priority. In-fact, AL approaches may be used in the context of few-shot learning to determine which samples may lead to the biggest performance gains, if used for adapting the task predictor.

The contributions of this work are summarised as follows: (1) We proposed a task-based AL metric with task-specific feedback from the targeted segmentation task; (2) We proposed to learn the prioritisation metric using meta-RL with adaptability over different imaging institutes and organ segmentation tasks; (3) We evaluated our proposed framework using real patient CT images and including segmentation tasks for anatomical structures such as liver, pancreas, spleen, liver vessels, gallbladder, adrenal glands (left and right), major vessels (aorta, vena cava and portal vein) and stomach; subsequently, the trained system was evaluated, for AL, on holdout tasks for liver vessels and kidneys for data from new institutes.

2. Methods

Pool sampling and stream sampling represent two typical cases for sampling the unlabelled data (Settles, 2009), in which the data that need annotation become available simultaneously or sequentially, respectively. The sampling method is usually determined by the application. For example, in many medical image segmentation tasks, there are often more unlabelled images than labelled data. Considering the pool sampling in this study, when an unlabelled dataset is available, the so-called *batch-mode* sampling (Settles, 2009) provides additional flexibility, with which the unlabelled images can be selected from and labelled in batches, hereinafter referred to as *AL batches*, to allow efficient and practical parallel processing and annotating, with the AL batch size being an additional hyperparameter.

We outline the task-based prioritisation metric formulation, in Section 2.2, where the segmentation performance is quantified for a pool of images. We then outline an algorithm to equip adaptability to the metric, in Section 2.3, such that it can be adapted to new datasets. Finally, we describe the usage of the learnt metric for the batch-mode AL, in Section 2.4, during which the prioritisation function is used to select samples to label and train a task predictor for the new dataset. A high-level summary of all stages is provided in Section 2.1.

2.1. Stage summary for the proposed method

We now provide a high-level overview of all the stages involved including controller training (Algo. 1), AL (Algo. 2) and evaluation using the holdout set.

The controller training stage involves the use of multiple MDP environments, each with their own dataset. The dataset within each MDP includes the controller-train and controller-validation sets. One set of interactions with a single MDP environment or trial occurs when a random batch of samples is sampled from the controller-train set and passed to the controller to obtain prioritisation scores. Samples are then selected for training the task predictor using these scores (as outlined in Algo. 1 using Reptile). Once trained, the task predictor is evaluated on the controller-weighted validation set (as described in Section 2.3.2 and Algo. 1). The performance metric obtained from this evaluation is then used as a reward signal to update the controller. Then, a new MDP is randomly sampled and the controller updates continue until convergence. The controller is an RNN and the internal state is reset after each trial. It is also noteworthy that the controller takes additional inputs of the action, raw reward, and termination flag at the previous time step, in addition to the observed current state (batch of samples form the controller-train set) which makes it adaptable rather than any update of the weights. This concept has been used and extensively validated in RNN-based meta-learning in previous works (Duan et al., 2016; Wang et al., 2017; Cotter and Conwell, 1990; Santoro et al., 2016; Younger et al., 1999; Prokhorov et al., 2002; Hochreiter et al., 2001). Since the controller is a function of the history leading up to a sample, if the history up to the sample is modified, the controller output may be modified, this is what makes the controller adaptable with fixed weights. This adaptability allows the controller to be adapted for new structures or even to adapt predictions for the same structure. As a concrete example, if a high-priority sample (as determined by the controller) is used for learning in AL and it impacts the performance negatively (as measured by the reward), then this becomes a part of the historic sequential data that is accessible to the controller. Based on this historic data, the controller can then modify its future predictions to prioritise any subsequent similar samples lower, since the meta-RL controller is a function of the history leading up to a sample rather than simply a function of the current sample, as in traditional RL.

For the AL stage, the controller weights are fixed after controller training and the internal state is reset before AL starts. Prior to AL, β^0 samples are randomly selected from a pool of unlabelled samples to initialise. These are labelled and split into support-train and support-validation portions. The support-train portion is used to updated the task predictor and the support validation portion is used to compute the reward which will be used as the controller input for the first AL iteration. The remaining samples in the pool of unlabelled samples (or the query set) are passed to the controller to obtain priority scores. β highest priority samples are then selected and labelled. These labelled are then split into support-train and support-validation sets for the iteration and added to the respective support sets from the previous iteration, for their respective task predictor and reward updates. It should be noted that the reward in this case is used as controller input and not to update the controller weights as in the controller-training stage. The reward serves as a signal to adapt the RNN internal state which equips the system with adaptability without weight updates for the controller. The AL iterations continue until either exhausting the query set or convergence.

To evaluate the AL performance, we use the task-predictor trained after AL. A holdout set, not used during the controller-training or AL stages, is passed to the trained task predictor. The performance measure is computed by comparing the task predictor predicted labels to the ground truth labels, where the mean binary Dice score is reported in our work.

Code available at: www.github.com/s-sd/task-amenability/tree/v2

2.2. Preliminary: bi-level optimisation for task-based prioritisation

In order to capture the task-specific performance for learning prioritisation, we consider two functions: (1) a *task predictor* which performs the segmentation task; and (2) a *controller* which predicts the task-specific prioritisation.

Defining the image and label domains for the task as $\mathcal{X}$ and $\mathcal{Y}$, respectively, we can denote the image and joint image-label distributions as $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ with probability density functions $p(x)$ and $p(x, y)$, respectively. Here $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are the sampled image and corresponding label. We can then define our two functions as follows.

The task predictor is assumed to be a parametric function: $f(\cdot; w) : \mathcal{X} \to \mathcal{Y}$, which outputs a prediction for the task $y \in \mathcal{Y}$ given an input image $x \in \mathcal{X}$, with parameters w. The controller is also assumed to be a parametric function: $h(\cdot; \theta) : \mathcal{X} \to [0, 1]$, which outputs a task-specific prioritisation score given an image sample x, with parameters θ. To optimise the task predictor, a loss function $L_f : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, which measures task performance, weighted by controller outputs $h(x; \theta)$, may be minimised:

$$\min_{w} \mathbb{E}_{(x,y)\sim\mathcal{P}_{XY}}[L_f(f(x; w), y)h(x; \theta)] \tag{1}$$

where, by weighting the loss using the controller outputs, the controller is incentivised to assign low scores to samples with high loss values.

The controller may be optimised by minimising a metric function $L_h : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ on the validation set, weighted by the controller outputs for the validation set $h(x; \theta)$:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{P}_{XY}}[L_h(f(x; w), y)h(x; \theta)], \tag{2}$$

$$\text{s.t.} \quad \mathbb{E}_{x\sim\mathcal{P}_X}[h(x; \theta)] \geq C > 0. \tag{3}$$

where, higher metric function values tend the controller towards lower output values due to the weighted sum being minimised. The constraint, C, prevents $h \equiv 0$ as a trivial solution.

We can thus pose the following bi-level optimisation:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{P}_{XY}}[L_h(f(x; w^*), y)h(x; \theta)], \tag{4}$$

$$\text{s.t.} \quad w^* = \arg\min_{w} \mathbb{E}_{(x,y)\sim\mathcal{P}_{XY}}[L_f(f(x; w), y)h(x; \theta)], \tag{5}$$

$$\mathbb{E}_{x\sim\mathcal{P}_X}[h(x; \theta)] \geq C > 0. \tag{6}$$

Replacing the above functions weighted by the prioritisation scores with functions that sample only the selected images, retaining the equal expected function values, the optimisation problem becomes:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{P}^h_{XY}}[L_h(f(x; w^*), y)], \tag{7}$$

$$\text{s.t.} \quad w^* = \arg\min_{w} \mathbb{E}_{(x,y)\sim\mathcal{P}^h_{XY}}[L_f(f(x; w), y)], \tag{8}$$

$$\mathbb{E}_{x\sim\mathcal{P}^h_X}[1] \geq C > 0. \tag{9}$$

where x and (x, y) to be sampled from the controller-selected or -sampled distributions $\mathcal{P}^h_X$ and $\mathcal{P}^h_{XY}$, with probability density functions $p^h(x) \propto p(x)h(x; \theta)$ and $p^h(x, y) \propto p(x, y)h(x; \theta)$, respectively.

RL algorithms, previously proposed for task-specific image quality assessment (Saeed et al., 2021a) or data valuation (Yoon et al., 2020), are adapted to optimise the task-based prioritisation metric. Building on this formulation, this work proposes an extension as well as a generalisation of such RL algorithms, a meta-RL approach for multiple datasets, as described in the remainder sections.

Once the controller is trained, it serves as a prioritisation function during AL. It is useful to clarify that, although the controller is denoted for individual samples for the simplicity in notation, its implementation includes a recurrent neural network (RNN), such that the episodic controller training (Algo. 1) and the batch-mode sampling in AL stage (Algo. 2) enable learning and inferring any potentially useful sequential information, respectively. Such RNN-embedded RL agent has also been adopted in other meta-RL approaches (Duan et al., 2016; Wang et al., 2017; Botvinick et al., 2019; Robles and Vanschoren, 2019).

2.3. Controller training stage: learning the adaptable prioritisation metric

In this work, the adaptability over different segmentation problems includes multi-organ adaptability as well as adaptability over different institutes, using the meta-RL-based training scheme described as follows.

2.3.1. Markov decision process environment

The above-outlined bi-level minimisation problem is modelled as a finite-horizon Markov decision process (MDP), where the controller interacts with an environment containing the task predictor and a specific set of data for training. The MDP-contained data are drawn from the distribution $\mathcal{P}_{XY}$, a joint image-label distribution, defined as $\mathcal{P}_{XY} = \mathcal{P}_X \mathcal{P}_{Y|X}$, with the task predictor as $f(\cdot; w)$. A train set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N$ is sampled from the distribution $\mathcal{P}_{XY}$, where N is the train set size. The observed state of the environment $s_t = (f(\cdot; w_t), \mathcal{B}_t)$, at time-step t, is composed of the task predictor $f(\cdot; w)$ and a mini-batch of b samples $\mathcal{B}_t = \{(x_i, y_i)\}_{i=1}^b$ from the train set.

2.3.2. Learning the metric using reinforcement learning

Reinforcement learning learns the prioritisation metric, by optimising the weights θ for the controller function $h(x; \theta)$. We define $\mathcal{S}$ as the state space and $\mathcal{A}$ as the continuous action space. $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ denotes the state transition distribution conditioned on state–actions, e.g. $p(s_{t+1}|s_t, a_t)$ represents the probability of the next state $s_{t+1} \in \mathcal{S}$ given the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function such that $R_t = r(s_t, a_t)$ denotes the reward given current state s_t and action a_t.

The policy, $\pi(a_t|s_t) : \mathcal{S} \times \mathcal{A} \in [0, 1]$, represents the probability of performing the action a_t given the state s_t. This allows for the MDP interaction to be summarised as a 5-tuple $(\mathcal{S}, \mathcal{A}, p, r, \pi)$. A number of interactions between the controller and the environments leads to a trajectory over multiple time-steps $(s_1, a_1, R_1, s_2, a_2, R_2, \ldots, s_T, a_T, R_T)$. The goal in this reinforcement learning is to optimise the controller parameters which maximise a cumulative reward over a trajectory.

In our work, we use a cumulative reward, starting from time-step t, of the form: $Q^\pi(s_t, a_t) = \sum_{k=0}^{T} \gamma^k R_{t+k}$, where the discount factor $\gamma \in [0, 1]$ is used to discount future rewards. Here the policy is parameterised by θ and denoted as π_θ. The central optimisation problem then is to find the optimal policy parameters $\theta^* = \text{argmax}_\theta \mathbb{E}_{\pi_\theta}\left[Q^\pi(s_t, a_t)\right]$

Following Section 2.2, the controller outputs sampling probabilities $\{h(x_{i,t}, \theta)\}_{i=1}^b$ based on the input image batch. The action $a_t = \{a_{i,t}\}_{i=1}^b \in \{0, 1\}^b$ leads to a sample selection decision for task predictor training, selected if $a_{i,t} = 1$. With $a_{i,t} \sim \text{Bernoulli}(h(x_{i,t}; \theta))$, the policy $\pi_\theta(a_t|s_t)$ is defined as:

$$\log \pi_\theta(a_t|s_t) = \sum_{i=1}^{b} h(x_{i,t}; \theta)a_{i,t} + (1 - h(x_{i,t}; \theta)(1 - a_{i,t})) \tag{10}$$

The reward R_t is based on the validation set $\mathcal{D}_{\text{val}} = \{(x_j, y_j)\}_{j=1}^M$ from the same distribution as the train set $\mathcal{P}_{XY}$, where M is the validation set size. Here we consider the validation set to be randomly sampled from the dataset, which is why it is considered to be from the same distribution as the train set. Note that during AL, described in Section 2.4, the role of the validation set is performed using $\mathcal{B}_{\text{support-validation}}$, with further details e.g., how these samples are chosen, provided in Section 2.4. The performance for the validation set is denoted as $\{l_{j,t}\}_{j=1}^M = \{L_h(f(x_j; w_t), y_j)\}_{j=1}^M$ and is used to compute the un-clipped reward, which is weighted by the prioritisation scores h_j, $R_{t,\text{un-clipped}} = -\frac{1}{M}\sum_{j=1}^{M} l_{j,t}h_j$. It is then clipped using a moving average, $R_t = \alpha_R R_{t,\text{clip}} + (1 - \alpha_R)R_{t,\text{un-clipped}}$, where α_R is a hyperparameter empirically set to 0.9. The final reward is then computed as $R_t = R_{t,\text{un-clipped}} - R_{t,\text{clip}}$.

This reward definition over a validation set, consisting of multiple samples, encourages a prioritisation metric to be learnt which promotes generalisability to new samples; this, in effect, captures a form of representativeness. At the same time, the task-specific nature of the reward signal allows for informativeness to be captured in the computed reward.

2.3.3. A distribution of MDP environments

We denote a distribution of MDP environments as $\mathcal{P}_M$. An MDP environment sampled from this distribution is thus denoted as $M_k \sim \mathcal{P}_M$. During training the controller, these MDP environments are collectively considered as the "meta-train" environments. An MDP environment is denoted using M_k such that the joint image-label distribution and task predictor within the environment can be denoted as $\mathcal{P}_{XY,k}$ and $f_k(\cdot; w_k)$, respectively.

In this work, the task predictor is a single neural network-based function approximator, shared across the MDP environments, denoted as $f(\cdot; w)$ by omitting k. The task predictor network are "synced" each time an environment is sampled, by updating the predictor parameters using a gradient-based meta-learning update step, detailed in the following section.

2.3.4. Meta-RL to learn the adaptable prioritisation function

Meta-RL aims to maximise the expected return over a distribution of environments, such that the trained controller may effectively adapt to new MDP environments sampled from the distribution (Wang et al., 2017; Duan et al., 2016; Botvinick et al., 2019). The proposed episodic meta-RL training differs from RL in four aspects: (1) The controller is shared across multiple MDP environments sampled from $\mathcal{P}_M$; (2) Interaction of the controller with each MDP $M_k \sim \mathcal{P}_M$ takes place over multiple episodes and is referred to as a 'trial'; (3) The controller embeds a RNN with the internal memory shared across episodes within the same trial. The RNN memory state is reset each time a new MDP is sampled. This enables adaptability with fixed weights, since the controller becomes a function of the history leading up to a sample of sequential input data; (4) The action a_t, raw reward r_t, and termination flag d_t at the previous time step are passed to the controller as input, in addition to the observed current state s_{t+1}; the input denoted as τ_t encompasses these additional inputs. For per-sample controller operation, $r_t = R_t$ at the episode end and zero otherwise, i.e. a sparse reward (Duan et al., 2016; Wang et al., 2017).

An episode, as outlined in Algo. 1, encompasses an update of the task predictor using controller selected samples, followed by a reward computation and update of the controller.

In addition to an adaptable controller, the task predictor is shared between different environments and is updated using the Reptile update scheme (Nichol et al., 2018), in order to equip adaptability to the task predictor as well. Whilst alternative gradient-based meta-learning algorithm may also be applicable, the Reptile-based task predictor update is efficient. The Reptile update is formed of two steps: (1) Perform gradient descent for the task predictor $f(\cdot; w_t)$, starting with weights w_t and ending in weights $w_{t,\text{new}}$; (2) Update the task predictor weights $w_t \leftarrow w_t + \epsilon(w_{t,\text{new}} - w_t)$. Where ϵ is set as 1.0 initially and linearly annealed to 0.0 as trials progress (Nichol et al., 2018). We use adaptive moment estimation (Kingma and Ba, 2017) as the gradient descent algorithm. After each episode, the weights are synced for the task predictor, between different sampled environments. The Reptile update ensures that the task predictor also learns to be sample-efficient when used for AL. The first step of the update is a simple gradient descent on using a batch of samples, to compute $w_{t,\text{new}}$. Then in the second step of the update, the weights of the task predictor are updated using $w_t \leftarrow w_t + \epsilon(w_{t,\text{new}} - w_t)$, which is a weighted sum between the new weights $w_{t,\text{new}}$ and the old weights w_t.

Adaptation to a new MDP sampled from the distribution of MDPs, $M_a \sim \mathcal{P}_M$, is initiated by resetting the RNN internal state once at the start of the adaptation. The controller network weights are fixed, such that the adaptability comes from the RNN internal state updates rather than updates of the weights. This proposed scheme adapts the prioritisation metric to new datasets and, perhaps more importantly, enables a "pre-trainable" weight-fixed controller for the subsequent AL stage (Section 2.4).

During the adaptation, the controller is adapted using the validation set for reward computation. The reward on the previous time-step is passed as an input explicitly to the controller to aid the above discussed adaptability. The detailed steps for leaning the adaptable task-based prioritisation are summarised in Algo. 1 and illustrated in Fig. 1.

Algorithm 1: Learning adaptable task-based prioritisation using multiple environments

Data: Multiple MDPs $M_k \sim \mathcal{P}_M$.
Result: Controller $h(\cdot; \theta)$.

while *not converged* **do**
  Sample an MDP $M_k \sim \mathcal{P}_M$;
  Reset the internal state of controller h;
  for *Each episode in all episodes* **do**
    for $t \leftarrow 1$ **to** T **do**
      Sample a training mini-batch $\mathcal{B}_t = \{(x_{i,t}, y_{i,t})\}_{i=1}^{b}$;
      Compute selection probabilities $\{h_{i,t}\}_{i=1}^{b} = \{h(\tau_{i,t}; \theta_t)\}_{i=1}^{b}$;
      Sample actions $a_t = \{a_{i,t}\}_{i=1}^{b}$ w.r.t. $a_{i,t} \sim \text{Bernoulli}(h_{i,t})$;
      Select samples $\mathcal{B}_{t,\text{selected}}$ from $\mathcal{B}_t$;
      Update predictor $f(\cdot; w_t)$ with $\mathcal{B}_{t,\text{selected}}$;
      Compute reward R_t;
    end
    Collect one episode $\{\mathcal{B}_t, a_t, R_t\}_{t=1}^{T}$;
    Update controller $h(\cdot; \theta)$ using the RL algorithm described in Sec. 2.3;
  end
end

2.4. AL stage: using the pre-trained prioritisation function

The AL stage involves the use of a *meta-test environment*, $M_a \sim \mathcal{P}_M$, which contains images from the joint image-label domain $\mathcal{P}_{XY}$, and is illustrated in Fig. 2, with the algorithm summarised in Algo. 2. The entire dataset $\mathcal{D}$ is unlabelled in the initial meta-test environment and becomes labelled as AL iterates. The meta-train environments, however, do have labels. For brevity, it is not indexed by time steps, where more detailed data subsets are.

To initialise AL, we sample β^0 images randomly from the dataset $\mathcal{D} = \{x_i\}_{i=1}^{N+M}$, which consists of a pool of unlabelled samples, where sizes N and M are the intended sizes for the support-train[1] and support-validation sets, respectively, if the entire data pool is exhausted. These sampled β^0 images, labelled by an expert observer, form the initial support set $\mathcal{B}_{\text{support}} = \{x_i, y_i\}_{i=1}^{\beta^0}$. This support set is split into support-train set $\mathcal{B}_{\text{support-train}} = \{x_i, y_i\}_{i=1}^{\beta^0 \times \phi}$ and a support-validation set $\mathcal{B}_{\text{support-validation}} = \{x_i, y_i\}_{i=1}^{\beta^0 \times (1-\phi)}$, using a ratio of ϕ. The support-train and support-validation sets are used to update the task predictor $f(\cdot; w)$ until convergence and to compute the reward, R_c, to form $\tau_{c+1,\cdot}$ which is passed to the controller as an explicit input on the next iteration, where $c = 0$, respectively, in the initialisation step.

With the remainder $N + M - \beta^0$ samples in the pool, the first AL iteration ($c = 1$ counts the iterations) computes prioritisation scores using the fixed-weight pre-trained controller, $\{h(\tau_{i,c}; \theta^*)\}_{i=1}^{N+M-\beta^0}$ where θ^* denotes pre-trained parameters, with only the RNN internal state adaptable. Among these unlabelled images $\{x_i\}_{i=1}^{N+M-\beta^0}$, the β samples that are scored the highest by the prioritisation function are selected and labelled to form the support set at the iteration, $\mathcal{B}_{c,\text{support}} = \{x_i, y_i\}_{i=1}^{\beta}$, which is then further split to form the current support-train set $\mathcal{B}_{c,\text{support-train}} = \{x_i, y_i\}_{i=1}^{(\beta^0+\beta)\times(\phi)}$ and support-validation set $\mathcal{B}_{c,\text{support-validation}} = \{x_i, y_i\}_{i=1}^{(\beta^0+\beta)\times(1-\phi)}$, and added into respective support sets i.e. to $\mathcal{B}_{\text{support-train}}$ and $\mathcal{B}_{\text{support-validation}}$. The task predictor

[1] The "support-" prefix indicates the available datasets during AL stage and are processed by the weight-fixed controller. Readers familiar with meta-learning terminology may notice the absence of the "query" set, however, in our formulation, query set may be thought of as the entire pool of data that are available for AL, before any data annotation takes place, while increasing annotated samples are forming the growing support sets.

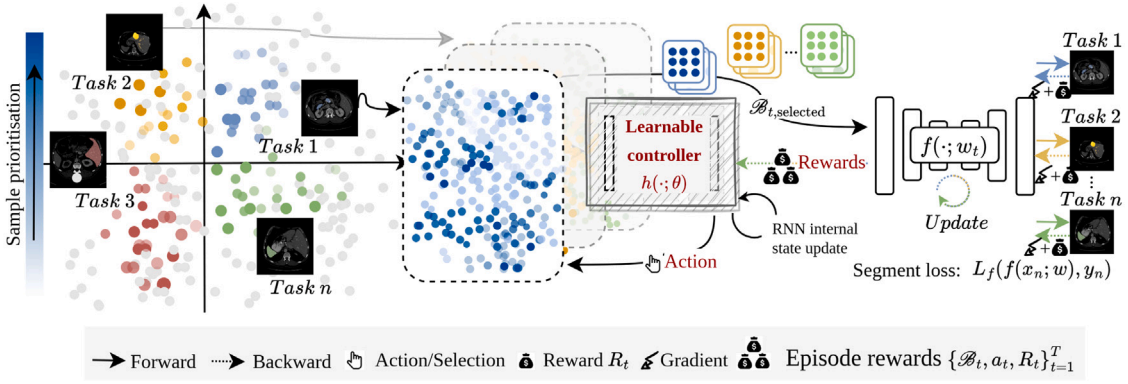


Fig. 1. Learning an adaptable prioritisation function. Learning happens across multiple environments each with a different segmentation task and data contained within the environment. Tasks are randomly sampled on each iteration for controller-environment interactions. Once an environment is sampled, a minibatch $\mathcal{B}_t$ is sampled from the data within the environment, which is passed to the controller in order to generate prioritisation scores/ selection probabilities for the samples. These are used to select samples (which is the controller action) to form $\mathcal{B}_{t,\text{selected}}$. $\mathcal{B}_{t,\text{selected}}$ is then used to update the task predictor. The reward used to train the controller comes from performance measured over the validation set $R_{t,\text{un-clipped}} = -\frac{1}{M}\sum_{j=1}^{M} l_{j,t}h_j$. Another environment is then randomly sampled and the whole process is repeated. Eventually, the controller learns to prioritise samples based on the task-specific feedback from the reward signal.

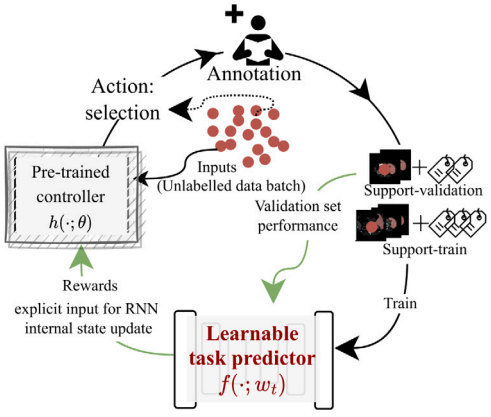


Fig. 2. AL. Pre-trained controller (fixed weights), acting as a prioritisation function, outputs prioritisation scores. Highest priority samples are annotated by a human observer and form the support-train and support-validation sets, used to update the task predictor and internal state of the RNN-based controller for adaptation (using reward on the support-validation set as an explicit input), respectively.

$f(\cdot;w)$ and the reward R_c are updated using the support-train and support-validation sets, respectively, as described in the initialisation step.

Thus in each subsequent cth iteration, $N + M - \beta^0 - (\beta \times (c-1))$ images are available for prioritising and $\beta^0 + (\beta \times c)$ are labelled. The process is then repeated until the pool of unlabelled $N + M$ images is exhausted or until AL convergence is reached.[2] The number of samples for initialising AL β^0 may be different from β, since initialisation may require more samples compared to a single AL iteration (Budd et al., 2021).

3. Experiments

In this section, we describe experiments to assess the proposed controller, by evaluating the segmentation performance for structures such as kidney, liver vessels and liver tumours on abdominal CT images, as example anatomical structures for surgical planning interest. The clinical datasets used for controller training are described in Section 3.1; The AL scenarios and the datasets are described in Section 3.2; The implementation is detailed in Section 3.3; And the remainder of Section 3.4 describes the alternative prioritisation metrics for ablation studies.

3.1. Controller training datasets

The training data for the adaptable model comes from multiple institutes, summarised in Table 1. The datasets in the table which have a cited reference as a source, are open-source datasets not requiring approval for usage; further ethical approval details for original acquisition may be found within the citations. Approval details for other datasets are provided in Section 3.2. The datasets include organ segmentation tasks for the liver, pancreas, spleen, liver vessels, gallbladder, adrenal gland (left and right), major vessels (aorta, vena cava, portal and splenic veins), stomach and kidneys. Here, the 'controller-' prefix indicates the use of the dataset during the controller training as outlined in Algo. 1. Two of the sets used for AL, i.e. those not used during controller training, are in-house datasets that come from UCL (University College London). It is noteworthy that the choice of annotation types were motivated by the intervention planning such as laparoscopic liver resection of interest in this study, as well as by the availability of the open data sets, which may further assist the reproducibility of the presented experimental results. As discussed in Section 1, combination of the training and support class types varies between applications and testing the ability to reduce the reliance on specific classes to be annotated for a large amount of training data is one of the focuses of the proposed cross-institute AL method. Summarised in Table 1, there are datasets used for controller training, this is the stage were our adaptable prioritisation metric is learnt and there are datasets used for AL, which are datasets that we use for evaluation of our proposed AL framework, these datasets were not used for learning the adaptable prioritisation metric and remain unseen by the controller or task predictor before AL.

3.2. Active learning tasks

The below described *meta-test* datasets were curated to represent the AL scenarios found in clinical practice. These scenarios include the segmentation of ROIs that are from a different institute and/or completely unseen in controller-training. They are a mixture of openly

[2] AL convergence is different from the model convergence within each AL iteration, and refers to the AL system converging at the highest performance value across multiple AL iterations. In the context of AL with respect to the predictor, an AL iteration is complete when the learning system has converged to a performance value, given the data available at that particular instance.

Algorithm 2: AL using an adaptable prioritisation function (fixed controller weights)

Data: MDP environment over which AL is to be conducted $M_a \sim \mathcal{P}_M$.
Result: Task predictor $f(\cdot; w)$.

Reset the internal state of controller h;
From the dataset $\mathcal{D} = \{x_i\}_{i=1}^{N+M}$, randomly sample β^0 images and label them to form $\mathcal{B}_{\text{support}} = \{x_i, y_i\}_{i=1}^{\beta^0}$;
Split $\mathcal{B}_{\text{support}}$ into support-train and support-validation portions $\mathcal{B}_{\text{support-train}} = \{x_i, y_i\}_{i=1}^{\beta^0 \times (\phi)}$ and $\mathcal{B}_{\text{support-validation}} = \{x_i, y_i\}_{i=1}^{\beta^0 \times (1-\phi)}$;
Update the task predictor $f(\cdot; w)$ using $\mathcal{B}_{\text{support-train}}$;
Compute the reward R_c used to form $\tau_{c+1,\cdot}$ where $c = 0$, using $\mathcal{B}_{\text{support-validation}}$;
for *each AL iteration* c *in all AL iterations* **do**
  while *not converged* **do**
    Compute prioritisation scores for the remaining samples in the dataset $\{h(\tau_{i,c}; \theta^*)\}_{i=1}^{N+M-\beta^0-(\beta \times (c-1))}$;
    Select β high priority samples from the remaining samples in $\mathcal{D}$ i.e. from $\{x_i\}_{i=1}^{N+M-\beta^0-(\beta \times (c-1))}$, label them and use them to form $\mathcal{B}_{c,\text{support}} = \{x_i, y_i\}_{i=1}^{\beta}$;
    Split $\mathcal{B}_{c,\text{support}}$ into support-train and support-validation portions $\mathcal{B}_{c,\text{support-train}} = \{x_i, y_i\}_{i=1}^{\beta \times (\phi)}$ and $\mathcal{B}_{c,\text{support-validation}} = \{x_i, y_i\}_{i=1}^{\beta \times (1-\phi)}$ and add them to $\mathcal{B}_{\text{support-train}}$ and $\mathcal{B}_{\text{support-validation}}$;
    Update predictor $f(\cdot; w_c)$ using $\mathcal{B}_{\text{support-train}}$;
    Compute reward R_c using $\mathcal{B}_{\text{support-validation}}$ and use it to form $\tau_{c+1,\cdot}$;
  end
end

Table 1
Sources of training datasets used to form different environments. For the 'Role', 'Training' means that two sets controller-train and controller-validation were created from the data; and 'AL' means that the data was used to form the support-train and support-validation sets and another set called the holdout set, used to evaluate the AL performance.

Source	Properties mm (slice thickness; resolution)	Structure	Samples	Role
Bilic et al. (2019), Antonelli et al. (2021)	0.8-4.0; 0.69–0.85	Liver	131	Training
Antonelli et al. (2021)	NA; NA	Pancreas	281	Training
Antonelli et al. (2021)	NA; NA	Spleen	41	Training
Antonelli et al. (2021)	NA; NA	Liver Vessels	303	Training
Synapse (2013)	2.5-5.0; 0.54–0.98	Gallbladder	30	Training
Synapse (2013)	2.5-5.0; 0.54–0.98	Adrenal Gland	30	Training
Synapse (2013)	2.5-5.0; 0.54–0.98	Major Vessels	30	Training
Synapse (2013)	2.5-5.0; 0.54–0.98	Stomach	30	Training
Rister et al. (2020)	NA ; 0.56–1.00	Kidneys	119	AL
UCL	NA; NA	Kidneys	35	AL
UCL	0.8-0.8 ; 0.12–0.12	Liver Vessels	9	AL
Bilic et al. (2023)	0.45-6.00; 0.56–1.00	Liver Tumour	131	AL

accessible datasets from challenges and real-world clinical datasets from our local hospitals, as described in Table 1. In particular, the UCL data for vessel segmentation and kidney segmentation, Experiment (c) and (b), were acquired from patients undergoing laparoscopic liver resection surgery and interventional radiology for the kidneys, respectively also discussed in Section 1. In these surgical and interventional applications, the pre-operative data with the same acquisition protocols are inherently scarce, due to the prevalence of the surgical procedures at the local hospitals. Together with the other experiment with a much larger dataset, these tasks were designed to reflect the benefits, from the proposed prioritisation function, for scenarios with variable unlabelled data availability. It should be noted that the ϕ parameter is simply the dataset split ratio of number of samples in support-train:support-validation. This is set empirically depending on the dataset size.

Cross-institute-and-organ kidney segmentation. Kidney segmentation data from Rister et al. (2020) is used as the first dataset for evaluation. This dataset contained 119 3D abdominal CT volumes with kidneys segmented manually for each volume. The data was split into 95, 12 and 12 samples in support-train, support-validation and holdout sets ($\phi = 0.89$).

Cross-institute kidney segmentation. A second kidney segmentation task was used as another environment for evaluation, this dataset comprised of 35 3D abdominal CT scans from patients who presented with renal cancer and were undergoing renal cryoablation. Approval from the local clinical governance committee was obtained prior to data collection. Since this dataset is formed of CT scans for interventional use, different imaging protocols have been used compared to other sets used for training. The kidneys were segmented manually by a trained biomedical engineering researcher. The data was split into 21, 7 and 7 samples in the support-train, support-validation and holdout sets ($\phi = 0.75$).

Cross-institute liver vessel segmentation. The third task used for evaluation is liver vessel segmentation. Nine 3D abdominal CT volumes (with liver vessels segmented using a commercial service (Visible-Patient, 2022)) were acquired in accordance with ethical standards of the institutional and/or national research committee and the 1964 Helsinki declaration (and amendments), under the study [REC=14/LO/1264] [IRAS=158321]. Data was split into 3, 3 and 3 samples in support-train, support-validation and holdout sets ($\phi = 0.5$).

Cross-institute-and-structure liver tumour segmentation. Liver tumour segmentation data from Bilic et al. (2023) is used for AL with 83, 24, and 24 samples in support-train, support-validation and holdout sets ($\phi = 0.78$).

3.3. Algorithm implementation

A 3D U-Net (Çiçek et al., 2016) was used as the task predictor shared between environments. Dice loss (i.e. 1 − Dice) was used for training across all environments. The U-Net was formed of 4 down-sampling and 4 up-sampling layers where down-sampling modules consisted of two convolutional layers with batch normalisation and ReLU activation, and a max-pooling operation and de-convolution layers used in up-sampling modules. Each convolution layer doubles the number of channels where the number of channels for the first layer is 32. Encoding and decoding parts were connected using skip connections.

The algorithm used for training the controller was Proximal Policy Optimisation (PPO) (Schulman et al., 2017). The actor and critic networks used in the PPO algorithm passes the image inputs via a 3-layered convolutional encoder which then feed into 3 fully connected layers, which embed an RNN. The reward used to train the controller was based on the Dice metric computed on the validation set. Using a single Nvidia Tesla V100 GPU, the controller training time was approximately 96 h.

In this work AL experiments are set up as outlined in Section 2.4. First, β^0 samples are randomly chosen and labelled by an expert. Subsequently, β additional samples are chosen on each new AL iteration, based on the prioritisation method used, and are labelled and used for further training together with samples from the previous AL iterations.

3.4. Compared methods and ablation studies

We use two prioritisation schemes in this work: (1) the proposed prioritisation scheme, which is the AL based adaptable prioritisation; and (2) a random prioritisation. We also use Monte-Carlo (MC) Dropout-based prioritisation to compare with our proposed prioritisation scheme and highlight this where comparisons are made. We compare the performance of the learning systems using t-tests and specify the AL iteration at which the comparison is being made, where appropriate.

To evaluate the efficacy of the prioritisation metric, we perform an ablation study where the proposed controller-based adaptable prioritisation scheme is ablated from the proposed framework. This means that while the task predictor is the adaptable Reptile-based version, the prioritisation using the controller is replaced with random prioritisation or using an alternative prioritisation scheme. Results presented in Fig. 3(d).

The reported performance is computed over a holdout set which is not used for training. The Dice score is reported for the learning systems over the holdout set. The settings for β^0 and β are specified in Section 4, where appropriate. Note that if $\beta \times (1 - \phi) < 1.0$ then samples need not be added to the support-validation set on every AL iteration.

4. Results

Cross-institute and cross-organ adaptability for kidney segmentation. As presented in Fig. 3(a), the proposed AL metric leads to faster convergence compared to random baseline with convergence reached near $c = 6$ for the proposed prioritisation scheme compared to $c = 12$ for the random prioritisation. It should be noted, however, that convergence is reached at a similar Dice score between these two ($p = 0.09$, t-test at α=0.05). While no difference was found between the two, both at $c = 12$ ($p = 0.11$), statistically significant difference was observed ($p < 0.001$) between the random prioritisation with the proposed prioritisation scheme both at $c = 6$.

Cross-institute adaptability for kidney segmentation. For the second kidney segmentation task, which demonstrates cross-institute-and-protocol adaptability, the plot of segmentation performance against AL iteration number is presented in Fig. 3(b). It appears as though convergence is not reached for either of the prioritisation schemes, however, at $c = 6$ we observed an improvement in the Dice score for the proposed prioritisation scheme, with statistical significance ($p < 0.001$). However, we observed higher Dice score for the proposed prioritisation metric for all values of c, with statistical significance ($p < 0.001$ for all).

Table 2
AL performance (Dice) measured over the holdout set for different prioritisation methods. Dice scores over 88.0 are indicated in bold, which are approximately considered as reached convergence, whereas the proposed method requires the least active learning iterations (c=6), 46%–67%, compared to other methods (c=13, 10 and 9). It is important to note that improvement at the converged performance neither is expected, nor a criterion for assessing AL algorithms.

c	Proposed	Random	Random (adaptable task predictor)	MC Dropout (adaptable task predictor)
1	60.89 ± 2.12	40.51 ± 1.67	60.89 ± 2.12	60.89 ± 2.12
2	67.87 ± 1.87	45.78 ± 2.45	63.23 ± 2.46	64.10 ± 2.34
3	79.76 ± 1.74	47.23 ± 2.98	67.38 ± 2.16	65.29 ± 3.11
4	86.43 ± 3.21	52.54 ± 2.74	72.52 ± 1.87	69.89 ± 2.87
5	86.72 ± 1.90	54.23 ± 2.63	76.22 ± 1.82	75.68 ± 1.98
6	**88.98 ± 2.43**	60.79 ± 1.99	78.89 ± 2.79	80.32 ± 2.34
7	**89.12 ± 2.76**	63.23 ± 3.10	79.32 ± 3.11	85.09 ± 2.78
8	**90.27 ± 1.98**	67.10 ± 2.72	83.41 ± 1.83	87.70 ± 2.84
9	**89.59 ± 2.33**	71.96 ± 1.91	86.92 ± 2.49	**89.93 ± 2.74**
10	**91.32 ± 2.64**	74.10 ± 1.73	**91.43 ± 2.85**	**90.12 ± 1.79**
11	**90.44 ± 2.83**	82.18 ± 2.34	**90.27 ± 2.60**	**91.21 ± 2.45**
12	**89.33 ± 1.85**	84.01 ± 2.38	**89.98 ± 2.46**	**89.87 ± 2.58**
13	**89.21 ± 1.96**	**88.10 ± 2.78**	**90.71 ± 3.10**	**89.21 ± 2.54**
14	**90.43 ± 2.14**	**89.43 ± 2.36**	**90.21 ± 1.84**	**90.81 ± 2.39**
15	**91.01 ± 2.87**	**90.53 ± 2.47**	**89.42 ± 1.94**	**91.41 ± 2.91**
16	**90.21 ± 2.32**	**91.24 ± 2.88**	**90.11 ± 2.39**	**90.87 ± 2.87**
17	**90.65 ± 1.71**	**91.21 ± 2.67**	**90.47 ± 2.98**	**89.10 ± 2.71**
18	**91.21 ± 2.13**	**90.50 ± 2.43**	**90.10 ± 2.35**	**88.98 ± 1.99**
19	**89.76 ± 2.47**	**89.83 ± 1.98**	**89.12 ± 2.73**	**89.78 ± 2.10**
20	**90.77 ± 2.82**	**89.24 ± 1.79**	**90.88 ± 2.28**	**91.13 ± 2.43**
21	**90.21 ± 1.89**	**89.10 ± 2.10**	**90.89 ± 1.98**	**90.10 ± 2.71**

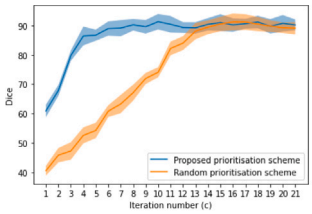
Cross-institute adaptability for liver vessel segmentation. For the liver vessel experiments, we present the AL results in Fig. 3(c). For this set of results, convergence may be considered inconclusive for either of the prioritisation schemes with the available data, however, we observe higher Dice score for the proposed prioritisation scheme, at $c = 5$, compared with random prioritisation, with statistical significance ($p < 0.001$). The proposed scheme results in higher performance compared to the random prioritisation baseline for all values of c, with statistical significance ($p < 0.001$ for all).

Ablation study for the kidney segmentation dataset. Comparing the ablated version with random prioritisation, a higher performance was seen from the proposed method from $c = 2$ to $c = 9$, with statistical significance ($p < 0.001$ for all). Similarly, the proposed method also outperformed the Monte-Carlo Dropout-based prioritisation for $c = 2$ to $c = 8$, with statistical significance ($p < 0.001$ for all). Interestingly, we found higher performance for the Monte-Carlo Dropout based scheme compared to random sampling for $c = 7$ and $c = 8$. The results are presented in Fig. 3(d). All models presented in this plot are initialised using the adaptable model and a different prioritisation scheme is used for each, subsequently. Additionally, numerical results are presented in Table 2.

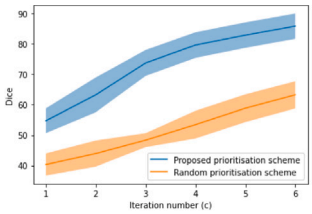
We also tested a variant where our RNN-controller was replaced with a vision transformer (ViT-Base) from Dosovitskiy et al. (2021). We observed scores over 88.0 Dice at c=5, which was the same as our RNN-controller.

Cross-institute and cross-structure adaptability for liver tumour segmentation. For the liver tumour segmentation task, we present results in Table 3 ($\beta^0 = 24$, $\beta = 4$)

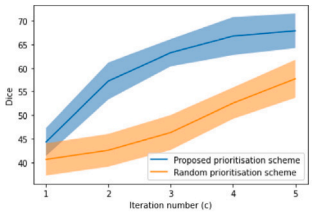
Exploring the impact of hyperparameters. Eight hyperparameter combinations (different values of β and β^0) were tested for the kidney segmentation task for the proposed active sampling. Consistent conclusions can be drawn with AL convergence being reached using nearly the same number of labelled samples as in our experiments outlined above. Detailed results are presented in Table 4. Fig. 4 illustrates final performance and number of samples required to reach convergence, relatively insensitive to varying hyperparameters.



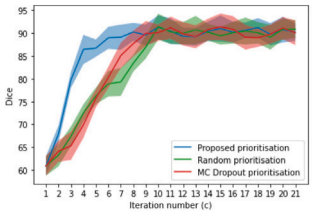
(a) Cross-institute and cross-organ adaptability for kidney segmentation. ($\beta^0 = 24, \beta = 4$)



(b) Cross-institute adaptability for kidney segmentation. ($\beta^0 = 8, \beta = 4$)

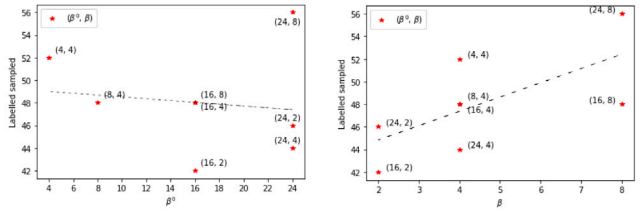


(c) Cross-institute adaptability for liver vessel segmentation. ($\beta^0 = 2, \beta = 1$)



(d) Ablation study for the kidney segmentation dataset. ($\beta^0 = 24, \beta = 4$)

Fig. 3. AL performance over the holdout set.



(a) β^0 (b) β

Fig. 4. Total number of labelled samples for convergence against hyperparameter, with fitted straight lines for reference.

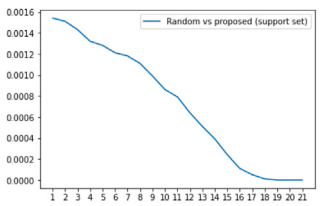
Table 3
Comparisons to other AL methods using the liver tumour segmentation task. Lower c indicates convergence in fewer iterations. Kohl et al. (2018) and Settles (2009) are the probabilistic U-Net and ensemble methods adapted from Czolbe et al. (2021). Bold indicates statistical significance for t-test.

Method	c at 74.0 Dice	Dice at c=21
Proposed	**6**	**76.92 ± 2.43**
Random	16	74.29 ± 2.53
MC Dropout	12	74.86 ± 2.10
Yang et al. (2017)	10	74.29 ± 2.37
Yoo and Kweon (2019)	9	74.66 ± 2.26
Kohl et al. (2018)	10	75.13 ± 1.95
Settles (2009)	9	74.52 ± 1.84

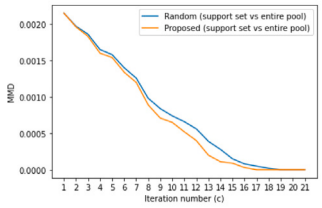
Analysis of the learnt selection strategies. To further analyse the selected samples at any given iteration, the Maximum Mean Discrepancy (MMD) was computed between $\mathcal{B}_{\text{support}}$ images selected by the two sampling strategies i.e. random and the proposed prioritisation, with the MMD value against the iteration number plotted in Fig. 5(a). MMD is also computed between the support set at iteration c and the entire pool of available data, plotted in Fig. 5(b). The random and proposed prioritisation follow a indistinguishable decreasing pattern, which may suggest the inability of MMD itself to differentiate or prioritise. Finally,

Table 4
Hyperparameters tested for the kidney task.

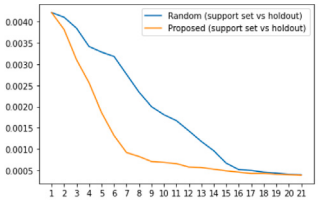
β^0	β	Dice	c at convergence	Labelled samples
24	4	88.98 ± 2.34	5	44
16	4	90.13 ± 1.83	8	48
8	4	91.02 ± 2.67	10	48
4	4	89.95 ± 1.94	12	52
24	2	88.87 ± 2.17	11	46
24	8	90.68 ± 1.75	4	56
16	2	89.40 ± 2.24	13	42
16	8	91.11 ± 1.99	4	48



(a) Support sets by random vs by the proposed



(b) Support set vs entire pool



(c) Support set vs holdout set

Fig. 5. MMD against iteration number for analysing the AL iteration convergence using the proposed method (see text in Section 4 for further details).

MMD computed between the support set at iteration c and the holdout set is presented in Fig. 5(c). In contrast, we observed faster decline in MMD for the proposed scheme in this case. The more observable difference may be due to the fact that the support is not a subset of the holdout and thus the divergence can be estimated more accurately using MMD. The difference itself may suggest a more representative selection learnt from the proposed method.

Qualitative analysis of results. Fig. 6 shows examples of segmented kidneys from the holdout dataset. Fig. 7 presents 3D models of liver vessels used as ground truth and corresponding prediction form the trained AL model. Fig. 8 compares examples selected by the proposed and random prioritisation schemes. It can be seen that randomly selected samples may include "low quality" or less-representative examples, for example, a sample with a missing label for one kidney (second) and a sample with disconnections in labels (third), while these cases are much fewer in samples selected by proposed prioritisation. Additionally, Fig. 9 compares samples deemed to be low priority by the proposed method and random prioritisation. It can be seen that the proposed method deems misaligned samples or samples with one kidney much smaller/larger than the other, to be low priority. This may be because these samples are rare in the dataset and thus offer low representativeness.

5. Discussion

Results from Section 4 show that the AL can effectively reduce the number of samples required to reach convergence, or a higher performance level, compared to random prioritisation. This means,

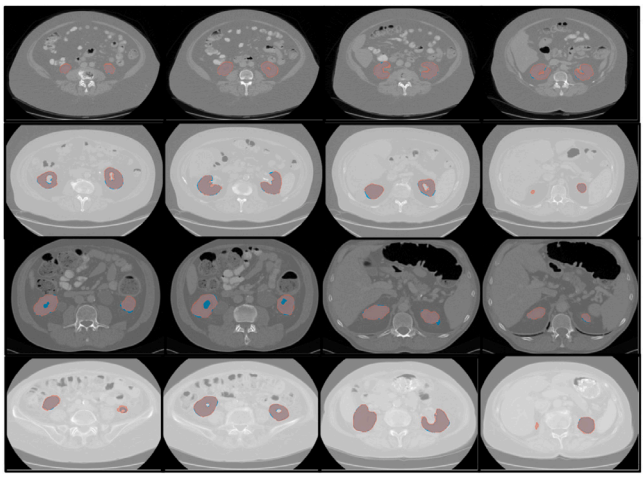


Fig. 6. Samples of kidney segmentation with ground truth (blue) and prediction (red), rows indicating same patients.

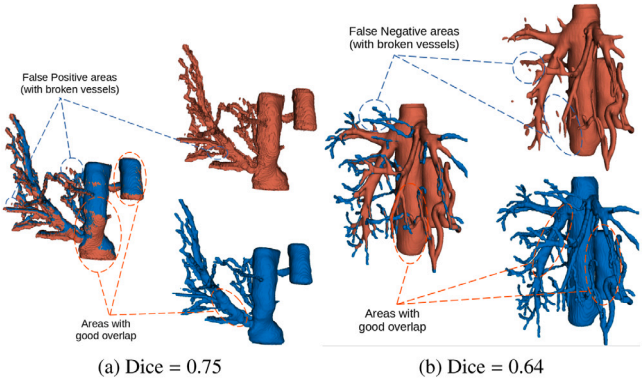


Fig. 7. 3D rendering of ground truth (blue) and overlaid prediction (red) for liver vessel task, from two holdout patients.

to achieve the same performance, experts would need to label fewer samples compared to annotating randomly sampled images. We demonstrate this using CT datasets for tasks such as: kidney segmentation, liver vessel segmentation and liver tumour segmentation. For kidney segmentation, the adaptable prioritisation metric yields converging segmentation accuracy using only 40%–60% of labels otherwise required with other prioritisation metrics or random sampling. For datasets with limited size, for kidney and liver vessel segmentation, the adaptable prioritisation metric offers a performance improvement of 22.6% and 10.2% in Dice score, respectively, compared to random prioritisation. This directly corresponds to savings in clinician time.

It is important to clarify that, based on the presented ablation results, the proposed adaptable prioritisation outperformed other schemes in the number of AL iterations to convergence. The performance at convergence, however, is comparable for all tested variants. Performance improvement is arguably not the goal of AL. However, the proposed adaptive AL has potential for a wider range of scenarios such as adaptable AL metrics across different tasks, across different observers and their labelling protocols, in addition to what are demonstrated in this paper, for novel structures and imaging institutes.

We demonstrated efficacy of the proposed approach, where we consider AL for CT scans from new institutes or for segmentation of unseen structures, evaluating under a variety of scenarios plausible in clinical settings. For example, labelling a relatively high object-to-background contrast anatomical structure and using this to train an

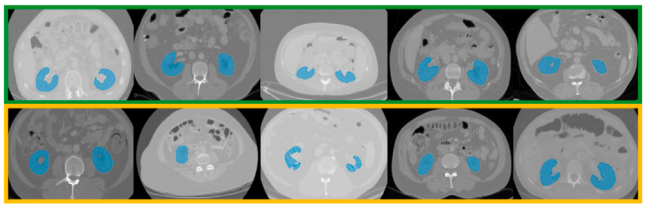


Fig. 8. Example samples selected by the proposed (green) and random prioritisation (yellow) for the kidney task ($c = 5$).

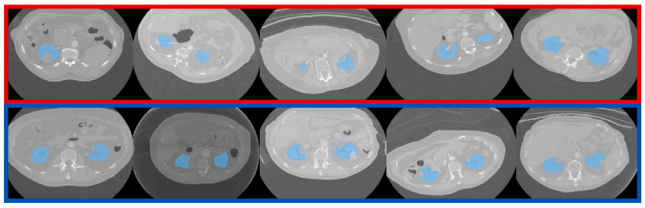


Fig. 9. Example samples deemed low-priority by the proposed (red) and random prioritisation (blue) for the kidney task ($c = 5$).

adaptable segmentation AL system, such as for liver segmentation, to aid the AL for a relatively low-contrast structure such as liver tumours may be a clinically plausible scenario. This is especially useful due to the potentially lower annotation burden of contouring anatomies vs pathologies, due to the relatively uniform appearances of anatomical structures vs the low object-to-background contrast, ambiguous boundaries and non-uniform appearances of pathologies such as tumours. We also demonstrated adapting the AL system to completely new datasets from unseen institutes, and for segmenting unseen structures, for kidney or liver vessel segmentation, which may be other clinically plausible scenarios for AL. Cross-image-modality adaptability or AL was, however, not evaluated due to additional challenges for high field-of-view scans such as magnetic resonance (MR) images. This is because MR images may be highly variable, dependent on the imaging sequences or protocols or institute-specific expertise and equipment. This could potentially be a direction for future investigation where cross-modality adaptability is investigated or multi-modality datasets are used for training, to aid the cross-modality adaptation.

It is also interesting to note that prioritisation for medical images may also indicate the clinical difficulty of the downstream tasks and therefore may aid the human annotator, e.g., challenging or low-priority samples may be passed to expert clinicians to segment rather than relying on automated segmentation. However, exploring this is beyond the scope of this work, where we focus on data efficiency.

6. Conclusion

This work introduces an adaptable AL metric learnt using a meta-RL approach which uses direct task-specific feedback for labelling prioritisation. The proposed method leads to faster convergence compared to random prioritisation and the widely used Monte-Carlo Dropout-based method. We demonstrated the applicability of the proposed approach on three segmentation tasks using multi-organ multi-institute CT data, in which the proposed prioritisation methods effectively provided approximately 40%–60% efficiency in using labelled data, over the alternative methods, to achieve the equivalent segmentation performance.

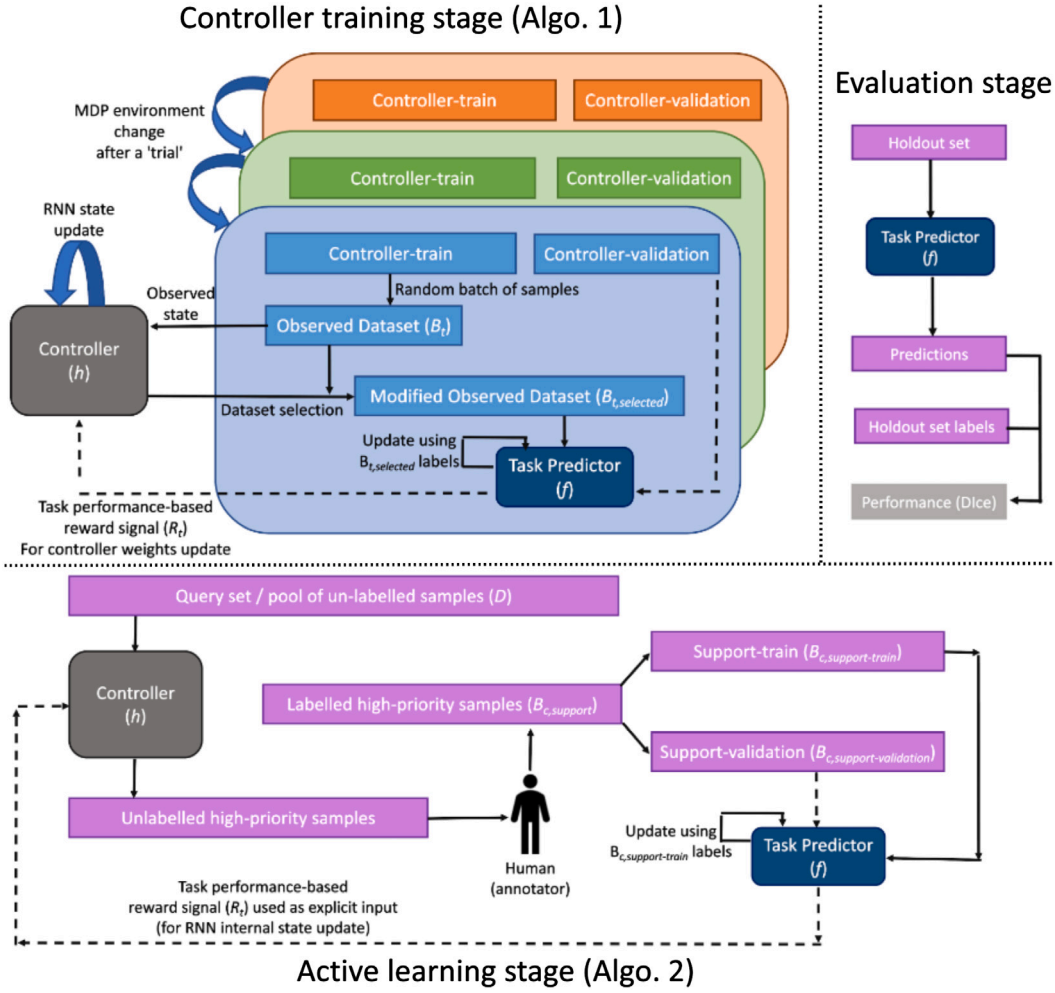


Fig. 10. A high-level overview of the controller training, AL and evaluation stages (solid arrows indicate the train set pathway and dashed arrows indicate the validation set pathway).

CRediT authorship contribution statement

Shaheer U. Saeed: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **João Ramalhinho:** Writing – review & editing, Funding acquisition, Data curation. **Mark Pinnock:** Writing – review & editing, Data curation. **Ziyi Shen:** Visualization. **Yunguan Fu:** Methodology, Conceptualization. **Nina Montaña-Brown:** Writing – review & editing, Data curation, Conceptualization. **Ester Bonmati:** Writing – review & editing, Data curation, Conceptualization. **Dean C. Barratt:** Writing – review & editing, Supervision, Funding acquisition. **Stephen P. Pereira:** Writing – review & editing, Project administration, Funding acquisition, Data curation, Conceptualization. **Brian Davidson:** Writing – review & editing, Supervision, Funding acquisition, Data curation, Conceptualization. **Matthew J. Clarkson:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Data curation, Conceptualization. **Yipeng Hu:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Funding acquisition, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Link to code in the paper but in-house patient data not made available due to ethics and approval limitations; other datasets are open-source and cited where appropriate.

Acknowledgements

This work is supported by the Engineering and Physical Sciences Research Council, United Kingdom (EPSRC) grant [EP/T029404/1]; Wellcome/EPSRC Centre for Interventional and Surgical Sciences [203145Z/16/Z]; the EPSRC CDT in i4health [EP/S021930/1]; the International Alliance for Cancer Early Detection, an alliance between Cancer Research UK [C28070/A30912; 73666/A31378], Canary Center at Stanford University, United States, the University of Cambridge, United Kingdom, OHSU Knight Cancer Institute, University College London and the University of Manchester, United Kingdom; and the National Institute for Health and Care Research (NIHR), United Kingdom under its Invention for Innovation (i4i) Programme [II-LA-1116-20005] (views expressed are those of the authors and not necessarily those of the NIHR or the Department of Health and Social Care).

Appendix

Overview of methodology

An overview of the methodology and the stages involved is presented in Fig. 10.

References

Antonelli, M., Reinke, A., Bakas, S., Farahani, K., AnnetteKopp-Schneider, Landman, B.A., Litjens, G., Menze, B., Ronneberger, O., Summers, R.M., van Ginneken, B., Bilello, M., Bilic, P., Christ, P.F., Do, R.K.G., Gollub, M.J., Heckers, S.H., Huisman, H., Jarnagin, W.R., McHugo, M.K., Napel, S., Pernicka, J.S.G., Rhode, K., Tobon-Gomez, C., Vorontsov, E., Huisman, H., Meakin, J.A., Ourselin, S., Wiesenfarth, M., Arbelaez, P., Bae, B., Chen, S., Daza, L., Feng, J., He, B., Isensee, F., Ji, Y., Jia, F., Kim, N., Kim, I., Merhof, D., Pai, A., Park, B., Perslev, M., Rezaiifar, R., Rippel, O., Sarasua, I., Shen, W., Son, J., Wachinger, C., Wang, L., Wang, Y., Xia, Y., Xu, D., Xu, Z., Zheng, Y., Simpson, A.L., Maier-Hein, L., Cardoso, M.J., 2021. The medical segmentation decathlon.

Bilic, P., Christ, P., Li, H.B., Vorontsov, E., Ben-Cohen, A., Kaissis, G., Szeskin, A., Jacobs, C., Mamani, G.E.H., Chartrand, G., et al., 2023. The liver tumor segmentation benchmark (lits). Med. Image Anal. 84, 102680.

Bilic, P., Christ, P.F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., Fu, C.W., Han, X., Heng, P.A., Hesser, J., Kadoury, S., Konopczynski, T., Le, M., Li, C., Li, X., Lipkovà, J., Lowengrub, J., Meine, H., Moltz, J.H., Pal, C., Piraud, M., Qi, X., Qi, J., Rempfler, M., Roth, K., Schenk, A., Sekuboyina, A., Vorontsov, E., Zhou, P., Hülsemeyer, C., Beetz, M., Ettlinger, F., Gruen, F., Kaissis, G., Lohöfer, F., Braren, R., Holch, J., Hofmann, F., Sommer, W., Heinemann, V., Jacobs, C., Mamani, G.E.H., van Ginneken, B., Chartrand, G., Tang, A., Drozdzal, M., Ben-Cohen, A., Klang, E., Amitai, M.M., Konen, E., Greenspan, H., Moreau, J., Hostettler, A., Soler, L., Vivanti, R., Szeskin, A., Lev-Cohain, N., Sosna, J., Joskowicz, L., Menze, B.H., 2019. The liver tumor segmentation benchmark (LiTS).

Botvinick, M., Ritter, S., Wang, J., Kurth-Nelson, Z., Blundell, C., Hassabis, D., 2019. Reinforcement learning, fast and slow. Trends Cogn. Sci. 23 (5), 408–422.

Budd, S., Robinson, E.C., Kainz, B., 2021. A survey on active learning and human-in-the-loop deep learning for medical image analysis. Med. Image Anal. 71, 102062.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3D U-net: Learning dense volumetric segmentation from sparse annotation. arXiv:1606.06650.

Cotter, N., Conwell, P., 1990. Fixed-weight networks can learn. In: 1990 IJCNN International Joint Conference on Neural Networks. pp. 553–559 vol.3.

Czolbe, S., Arnavaz, K., Krause, O., Feragen, A., 2021. Is segmentation uncertainty useful? In: International Conference on Information Processing in Medical Imaging. Springer, pp. 715–726.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.

Duan, Y., Schulman, J., Chen, X., Bartlett, P., Sutskever, I., Abbeel, P., 2016. RL^2: Fast reinforcement learning via slow reinforcement learning. arXiv:1611.02779.

Erickson, B.J., Korfiatis, P., Akkus, Z., Kline, T.L., 2017. Machine learning for medical imaging. Radiographics 37 (2), 505–515.

Fang, M., Li, Y., Cohn, T., 2017. Learning how to active learn: A deep reinforcement learning approach.

Feyjie, A.R., Azad, R., Pedersoli, M., Kauffman, C., Ayed, I.B., Dolz, J., 2020. Semi-supervised few-shot learning for medical image segmentation. arXiv preprint arXiv:2003.08462.

Fu, Y., Lei, Y., Wang, T., Curran, W.J., Liu, T., Yang, X., 2021. A review of deep learning based methods for medical image multi-organ segmentation. Phys. Med. 85, 107–122.

Fusaglia, M., Tinguely, P., Banz, V., Weber, S., Lu, H., 2016. A novel ultrasound-based registration for image-guided laparoscopic liver ablation. Surg. Innov. 23 (4), 397–406.

Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. PMLR, pp. 1050–1059.

Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., Davidson, B., Pereira, S.P., Clarkson, M.J., Barratt, D.C., 2018. Automatic multi-organ segmentation on abdominal CT with dense V-Networks. IEEE Trans. Med. Imaging 37 (8), 1822–1834.

Hochreiter, S., Younger, A., Conwell, P., 2001. Learning to learn using gradient descent. In: International Conference on Artificial Neural Networks. Springer, pp. 87–94.

Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization. arXiv:1412.6980.

Kohl, S., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J.R., Maier-Hein, K., Eslami, S., Jimenez Rezende, D., Ronneberger, O., 2018. A probabilistic u-net for segmentation of ambiguous images. Adv. Neural Inf. Process. Syst. 31.

Lee, J.-G., Jun, S., Cho, Y.-W., Lee, H., Kim, G.B., Seo, J.B., Kim, N., 2017. Deep learning in medical imaging: general overview. Korean J. Radiol. 18 (4), 570–584.

Lewis, D.D., Gale, W.A., 1994. A sequential algorithm for training text classifiers. In: Croft, B.W., van Rijsbergen, C.J. (Eds.), SIGIR '94. Springer London, London, pp. 3–12.

Li, Y., Fu, Y., Yang, Q., Min, Z., Yan, W., Huisman, H., Barratt, D., Prisacariu, V.A., Hu, Y., 2022. Few-shot image segmentation for cross-institution male pelvic organs using registration-assisted prototypical learning. In: 2022 IEEE 19th International Symposium on Biomedical Imaging. ISBI, IEEE, pp. 1–5.

Nichol, A., Achiam, J., Schulman, J., 2018. On first-order meta-learning algorithms. arXiv:1803.02999.

Ozdemir, F., Peng, Z., Tanner, C., Fuernstahl, P., Goksel, O., 2018. Active learning for segmentation by optimizing content information for maximal entropy. In: Stoyanov, D., Taylor, Z., Carneiro, G., Syeda-Mahmood, T., Martel, A., Maier-Hein, L., Tavares, J.a.M.R., Bradley, A., Papa, J.a.P., Belagiannis, V., Nascimento, J.C., Lu, Z., Conjeti, S., Moradi, M., Greenspan, H., Madabhushi, A. (Eds.), Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer International Publishing, Cham, pp. 183–191.

Pang, K., Dong, M., Wu, Y., Hospedales, T., 2018. Meta-learning transferable active learning policies by deep reinforcement learning.

Peng, Y., Zheng, H., Liang, P., Zhang, L., Zaman, F., Wu, X., Sonka, M., Chen, D.Z., 2022. KCB-Net: A 3D knee cartilage and bone segmentation network via sparse annotation. Med. Image Anal. 82, 102574.

Prokhorov, D., Feldkarnp, L., Tyukin, I., 2002. Adaptive behavior with fixed weights in RNN: an overview. In: Proceedings of the 2002 International Joint Conference on Neural Networks, Vol. 3. IJCNN'02, pp. 2018–2022.

Ramalhinho, J., Koo, B., Montaña-Brown, N., Saeed, S.U., Bonmati, E., Gurusamy, K., Pereira, S.P., Davidson, B., Hu, Y., Clarkson, M.J., 2022. Deep hashing for global registration of untracked 2D laparoscopic ultrasound to CT. Int. J. Comput. Assist. Radiol. Surg. 1–8.

Rister, B., Yi, D., Shivakumar, K., Nobashi, T., Rubin, D.L., 2020. CT-ORG, a new dataset for multiple organ segmentation in computed tomography. Sci. Data 7 (1), 1–9.

Robles, G., Vanschoren, J., 2019. Learning to reinforcement learn for neural architecture search. arXiv:1911.03769.

Saeed, S.U., Fu, Y., Baum, Z.M.C., Yang, Q., Rusu, M., Fan, R.E., Sonn, G.A., Barratt, D.C., Hu, Y., 2021a. Learning image quality assessment by reinforcing task amenable data selection. In: Feragen, A., Sommer, S., Schnabel, J., Nielsen, M. (Eds.), Information Processing in Medical Imaging. Springer International Publishing, Cham, pp. 755–766.

Saeed, S.U., Fu, Y., Stavrinides, V., Baum, Z.M.C., Yang, Q., Rusu, M., Fan, R.E., Sonn, G.A., Noble, J.A., Barratt, D.C., Hu, Y., 2021b. Adaptable image quality assessment using meta-reinforcement learning of task amenability. In: Noble, J.A., Aylward, S., Grimwood, A., Min, Z., Lee, S.L., Hu, Y. (Eds.), Simplifying Medical Ultrasound. Springer International Publishing, Cham, pp. 191–201.

Saeed, S.U., Fu, Y., Stavrinides, V., Baum, Z.M., Yang, Q., Rusu, M., Fan, R.E., Sonn, G.A., Noble, J.A., Barratt, D.C., Hu, Y., 2022a. Image quality assessment for machine learning tasks using meta-reinforcement learning. Med. Image Anal. 78, 102427.

Saeed, S.U., Yan, W., Fu, Y., Giganti, F., Yang, Q., Baum, Z., Rusu, M., Fan, R.E., Sonn, G.A., Emberton, M., et al., 2022b. Image quality assessment by overlapping task-specific and task-agnostic measures: application to prostate multiparametric MR images for cancer segmentation. arXiv preprint arXiv:2202.09798.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T., 2016. Meta-learning with memory-augmented neural networks. In: Balcan, M., Weinberger, K. (Eds.), Proceedings of the 33rd International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 48, PMLR, New York, New York, USA, pp. 1842–1850.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv:1707.06347.

Settles, B., 2009. Active Learning Literature Survey. University of Wisconsin-Madison Department of Computer Sciences.

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27 (3), 379–423.

Smailagic, A., Costa, P., Noh, H.Y., Walawalkar, D., Khandelwal, K., Galdran, A., Mirshekari, M., Fagert, J., Xu, S., Zhang, P., Campilho, A.J.C., 2018. Medal: Accurate and robust deep active learning for medical image analysis. In: 2018 17th IEEE International Conference on Machine Learning and Applications. ICMLA, pp. 481–488.

Synapse, 2013. Multi-atlas labeling beyond the cranial vault - workshop and challenge. http://dx.doi.org/10.7303/syn3193805, URL: https://www.synapse.org/#!Synapse:syn3193805/wiki/217789. (Online; accessed Jun 2022).

Visible-Patient, 2022. URL: www.visiblepatient.com. (Online; accessed Dec 2022).

Wang, J., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J., Munos, R., Blundell, C., Kumaran, D., Botvinick, M., 2017. Learning to reinforcement learn. arXiv:1611.05763.

Woodward, M., Finn, C., 2017. Active one-shot learning.

Yang, L., Zhang, Y., Chen, J., Zhang, S., Chen, D.Z., 2017. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (Eds.), Medical Image Computing and Computer Assisted Intervention. MICCAI 2017, Springer International Publishing, Cham, pp. 399–407.

Yoo, D., Kweon, I.S., 2019. Learning loss for active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 93–102.

Yoon, J., Arik, S., Pfister, T., 2020. Data valuation using reinforcement learning. arXiv:1909.11671.

Younger, A., Conwell, P., Cotter, N., 1999. Fixed-weight on-line learning. IEEE Trans. Neural Netw. 10 (2), 272–283.

Zhou, Z., Shin, J., Zhang, L., Gurudu, S., Gotway, M., Liang, J., 2017. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7340–7351.