

RLogist: Fast Observation Strategy on Whole-Slide Images with Deep Reinforcement Learning

Boxuan Zhao^{1,2,*}, Jun Zhang^{1,*}, Deheng Ye^{1,†}, Jian Cao², Xiao Han¹, Qiang Fu¹, Wei Yang¹

¹ Tencent AI Lab

² Shanghai Jiao Tong University

b-x-zhao@sjtu.edu.cn, junejzhang@tencent.com, dericye@tencent.com, cao-jian@sjtu.edu.cn,
haroldhan@tencent.com, leonfu@tencent.com, willyang@tencent.com

Abstract

Whole-slide images (WSI) in computational pathology have high resolution with gigapixel size, but are generally with sparse regions of interest, which leads to weak diagnostic relevance and data inefficiency for each area in the slide. Most of the existing methods rely on a multiple instance learning framework that requires densely sampling local patches at high magnification. The limitation is evident in the application stage as the heavy computation for extracting patch-level features is inevitable. In this paper, we develop RLogist, a benchmarking deep reinforcement learning (DRL) method for fast observation strategy on WSIs. Imitating the diagnostic logic of human pathologists, our RL agent learns how to find regions of observation value and obtain representative features across multiple resolution levels, without having to analyze each part of the WSI at the high magnification. We benchmark our method on two whole-slide level classification tasks, including detection of metastases in WSIs of lymph node sections, and subtyping of lung cancer. Experimental results demonstrate that RLogist achieves competitive classification performance compared to typical multiple instance learning algorithms, while having a significantly short observation path. In addition, the observation path given by RLogist provides good decision-making interpretability, and its ability of reading path navigation can potentially be used by pathologists for educational/assistive purposes. Our code is available at: <https://github.com/tencent-ailab/RLogist>.

Introduction

Deep learning methods have contributed much to the domain of medical image analysis, by solving image-/pixel-level classification and regression problems. However, whole-slide image (WSI) analysis remains to be challenging for deep learning, due to its uniqueness on the gigapixel size and the lack of pixel-level annotations, as illustrated in Figure 1. The task of WSI classification, such as cancer classification, lymph node metastasis detection, and immunohistochemical scoring, is formulated as a weakly supervised learning problem that can be solved by multiple instance learning (MIL).

Generally, all image patches from one WSI are regarded as instances to construct a bag, i.e., a collection of unlabeled instances, where all instances are negative if the WSI label is

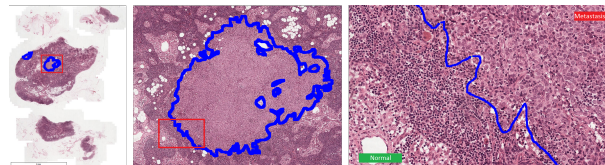


Figure 1: Metastatic and normal regions in H&E stained WSIs of lymph node sections. In most datasets, disease-positive tissue sections take only a small portion of the whole slide and pixel-level annotations are not available.

negative, and there exists at least one positive instance otherwise. MIL methods aggregate instance-level information to obtain bag-level representations for bag-level classification. Existing deep learning based MIL methods try to localize the diagnostic relevant areas by applying feature embedding non-parametric pooling (e.g., mean/max pooling), self-attention (Ilse, Tomczak, and Welling 2018; Lu et al. 2021), non-local attention with Transformer (Shao et al. 2021), etc, resulting in densely sampling for feature aggregation.

Current methods usually extract patch-level features for each instance based on a pretrained visual embedding, e.g., the training with ImageNet (Krizhevsky, Sutskever, and Hinton 2012), and self-supervised training using pathology data (Wang et al. 2021; Li, Li, and Eliceiri 2021). Such operation can be performed once for all and then cached in the training stage to reduce the burden of feature extraction, which alleviates the computational intensity. To date, most weakly-supervised studies focus on extracting informative instance (patch) features and aggregating these features.

However, during the inference stage, the drawback of having a heavy computational cost for extracting local patch features in an exhausted manner has often been neglected. With pretrained visual embeddings, e.g., ResNet50 (He et al. 2016), the feature extraction process usually requires dozens of minutes. Such a limitation becomes evident in clinical applications, as the MIL models require all instance features to perform the classification or regression. Therefore, it is necessary to develop a data-efficient method that can manage to avoid densely sampling all image patches in the high magnification and process informative features sparsely.

To this end, we aim to develop a fast observation strategy to avoid the aforementioned exhausting feature extraction. Motivated by the decision-making process of human

*Equal contribution; †Corresponding author
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

pathologists who sparsely analyze field of views at different scales by switching objective magnification powers of the microscope, it would be inspiring to have a fast observation strategy that selectively observes diagnostic-relevant regions without having to analyze each part of the WSI at a high magnification.

In this paper, we model the diagnostic decision-making process as a Markov Decision Process (MDP), where a human-imitating agent first looks for regions of observation value by scanning at low magnification and then choose potential diagnostic-relevant regions to observe in depth. Based on the idea of super-resolution, the agent is designed to continuously accumulate knowledge of what the scene of high magnification may look like for an unobserved low-magnification region by browsing across multiple resolution levels, i.e., updating the low-level representation of unobserved areas with observed region feature pairs at different resolution levels. Consequently, a quick slide-level classification decision is made by repeating the aforementioned pattern to generate a short observation pathway. Inspired by AI for gaming (Lin et al. 2022; Ye et al. 2020a,b), we train an agent to solve such a path-decision problem which is inherently related to optimal path selection with deep reinforcement learning (DRL).

To sum up, we make the following contributions:

- We develop RLogist, a DRL method for whole-slide image analysis, so as to avoid analyzing local image patches in a traditional exhaustive manner.
- Inspired by the decision-making process of a human, we implement the cross-magnification information fusion by conditional feature super-resolution. Benefiting from the conditional modeling, the high-magnification features of unobserved region could be updated according to the sparsely observed low-magnification and high-magnification feature pairs.
- We propose a well-defined environment for reinforcement learning algorithms, using the discrete action space to improve the model convergence performance by patching and the done-state reward to avoid local optima.
- Extensive experiments demonstrate that the performance of our RL agent is comparable to state-of-the-art methods while having a significantly shortened observation path.

Related Work

We develop a fast observation strategy for WSI analysis using DRL models. In this section, we review recent efforts on WSI analysis with MIL framework, as well as relevant works on DRL methods in medical image analysis.

WSI Analysis with Multiple Instance Learning

Weakly supervised learning with neural networks has been widely adopted for many classification tasks such as tumor detection, because the diagnostic areas that contributes to the slide-level label only occupies a relatively small proportion of the WSI (Campanella et al. 2019). Without detailed annotation of diagnostic-relevant regions, i.e., only slide-level labels are available, WSI classification is usually cast

as a multiple instance learning (MIL) problem (Carbonneau et al. 2018). Recent works have shown that an MIL classifier trained on large weakly-labeled WSI datasets even generalizes better than a fully-supervised classifier trained on pixel-level-annotated small lab datasets (Bulten et al. 2022; Campanella et al. 2019).

Conventional MIL models use handcrafted operators, such as mean-pooling or max-pooling to aggregate all feature embeddings of the patches in a whole slide (Pinheiro and Collobert 2015; Feng and Zhou 2017). To improve the data efficiency and classification performance, variant MIL attention based methods assign more accurate contribution weight of each instance by introducing neural network based aggregators with trainable parameters (Hashimoto et al. 2020; Zhao et al. 2020; Xie et al. 2020). Non-local attention mechanisms are introduced to model the relations of the instances in a dual-stream architecture with trainable distance measurement (Li, Li, and Eliceiri 2021). Transformer is also adopted in MIL to explore morphological and spatial information among different instance (Shao et al. 2021). Chen et al. (2021) proposed an aggregation operator with relevance-enhanced graph convolutional network to jointly analyze visual fields in a multi-scale manner for diagnostic predictions. In short, recent MIL methods mainly focus on assigning better contribution weights to patch instances by exploring and exploiting the correlated information between different instances or different resolution levels (Chen et al. 2022). Few of them focus on fast observation path planning.

DRL Methods in Medical Image Analysis

Deep reinforcement learning (DRL) augments the reinforcement learning framework, which learns a sequence of actions that maximizes the expected reward, with the representative power of deep neural networks (Li 2017; Silver et al. 2017; Zhao et al. 2021). Recent works have demonstrated the great potential of DRL in medicine and healthcare (Zhou et al. 2021). DRL methods are widely applied in parametric medical image analysis tasks including landmark detection, object/lesion detection, registration, and view plane localization. Alansary et al. (2019) evaluate different reinforcement learning agents with several training strategies for detecting anatomical landmarks in 3D images. Qaiser and Rajpoot (2019) treat IHC scoring as a sequential learning task to learn discriminative features and select informative regions within a cropped image (2048×2048) of a WSI. Note that although this method is designed for pathology image classification, it is not designed for whole-slide images. Because DRL can handle the non-differential metrics, it is widely used to solve optimization problems where conventional methods fall apart (Zhou et al. 2021). Such applications includes tuning hyperparameters for radiotherapy planning (Shen et al. 2018), selecting the right image augmentation selection for image classification (Cheng et al. 2019; Ye et al. 2020c; Yang et al. 2020; Wang et al. 2020), searching best neural architecture for segmentation (Furuta, Inoue, and Yamasaki 2019; Pineda et al. 2020), and avoiding poor images via a learned acquisition strategy (Zaech et al. 2020). DRL methods are also used in miscellaneous applications, such as surgical gesture segmentation (Liu and Jiang

2018), personalized mobile health intervention (Zhu et al. 2018), and computational model personalization (Abdi et al. 2020; Joos, Péan, and Goksel 2020).

However, realizing the full potential of DRL for medical imaging requires solving several challenges that includes defining an appropriate reward function, data availability, dynamic environment and etc (Zhou et al. 2021; Coronato et al. 2020). In addition, advanced DRL algorithms which have been proven to perform better and converge easier in many application fields, such as Proximal Policy Optimization (PPO) (Schulman et al. 2017) and Soft Actor-Critic (SAC) (Haarnoja et al. 2018), are still not widely used in computational pathology. In this paper, for the first time, we propose a generic deep reinforcement learning framework for observation and prediction strategy on WSIs with a precise definition of the state and action space, dealing well with dynamic environments and reward design issues.

Method

In this section, we depict the design of RLogist for fast observation strategy on whole-slide images. The overview architecture is shown in Figure 2. We first introduce the formulation of the observation process on WSIs, which is modeled as a Markov Decision Process (MDP), then present the reinforcement learning architecture and concrete descriptions of the components in our proposed model.

Problem Formulation

Given a WSI I , the task is to predict the slide-level label by scanning the image at low magnification (global visual perception) and observing potential diagnostic-relevant regions in depth at high magnification (local visual perception). The objective is to find the shortest possible observation path, while ensuring the slide-level prediction performance.

After tissue region segmentation, non-overlapping image patches with resolution $[l \times l]$ of multiple target magnification M are extracted from the WSI, annotated as $\{x_i^M\}_{i=1}^N \in \mathbb{R}^{N \times [l \times l]}$, where N is the number of patches varying from slide to slide. With a feature extractor f , each patch instance x_i can be described by a fixed d -dimensional vector representation $v_i = f(x_i) \in \mathbb{R}^{1 \times d}$. For image patches of the scanning-level magnification in a WSI, we refer to the collection of the extracted feature vectors as $\{v_i\}_{i=1}^N \in \mathbb{R}^{N \times d}$. The feature vectors of higher magnification patches belonging to each scanning-level magnification patch x_i are annotated as $\{v_j^i\}_{j=1}^K \in \mathbb{R}^{K \times d}$, where K is the number of higher magnification sub-patches. For example, an observation strategy with 5x scanning-level magnification and 20x observation-level magnification results in the sub-patch number $k = 16$.

Imitating the observation pathway of human pathologists on a whole-slide image, we model the decision-making process as a Markov decision process (MDP). At each time step t , the agent gets an observation $\{v_i^t\}_{i=1}^N$ of the state s_t by scanning at low magnification and then choose a potential diagnostic-relevant region x_i to observe in depth as action a_t . After observing sub-patches in region x_i to get local detailed features $\{v_j^i\}_{j=1}^K$, the process responds at the

Algorithm 1: The Agent-Environment Loop at Step t

Input: Current state s_t ;

Parameters: Policy parameter θ_t ; State transition function $f(s, a)$ represented by local feature updater f_{local} and unobserved region feature updater f_{global} ;

Output: Next state s_{t+1} and action a_t ;

- 1: Sample a_t from the distribution $A_t \sim \pi_{\theta_t}(\cdot|s_t)$
 - 2: **if** training **then**
 - 3: Update the policy π_{θ} with an RL algorithm
 - 4: **end if**
 - 5: Observe a region corresponding to a_t in the WSI and get a set of low-level feature embedding $\{v_j^{a_t}\}_{j=1}^K$
 - 6: Update the scanning-level feature of region a_t :
 $v'_{a_t} = f_{local}(\{v_j^{a_t}\}_{j=1}^K)$
 - 7: **for** scanning level region index i **in** $\{1, 2, \dots, N\} \setminus a_t$ **do**
 - 8: The unobserved region: $v'_i = f_{global}(v_i, (v_{a_t}, v'_{a_t}))$
 - 9: **end for**
 - 10: Transit to next state s_{t+1} with v'_i and v'_{a_t}
-

next time step t' by moving into a new state s' with updated scanning-level representative features $\{v_i^{t'}\}_{i=1}^N$. This step is more like a pathologist can infer unseen details through global visual perception and partial detailed field of views. In certain observation steps, the agent generates a prediction of the slide-level classification label and gets a trajectory reward R_{τ} , which is correlated with the cross-entropy of the output logit and the ground-truth. With a deep reinforcement model, we can train the agent to learn a parameterized policy π_{θ} optimizing the length of the observation path, i.e., to obtain a fast observation strategy.

In our task, there are several key problems that need to be solved. First, how observation actions affect the state transition process, i.e., updating scanning-level representative features $\{v_i^{t'}\}_{i=1}^N$. Second, how to handle the trade-off between the length of the observation path and the classification performance. We will elaborate our RLogist solution to these problems in the following subsections.

RLogist: Reinforcement Learning for Observation

In the proposed model, our key innovation is that the RL agent not only develops a discriminative capacity of diagnostic-relevant regions in the training stage, but also gradually improves the cognitive ability of unobserved regions by accumulating information about the observed regions in the inference stage. Based on the idea of super-resolution, the latter mechanism conditionally updates the scanning-level feature of unobserved regions $\{v_i\}_{i=1}^{N'}$ with the low-high feature pair of the new-observed region, which can improve the accuracy of target region selection and accelerate the observation process significantly. We now describe each components of the RLogist in detail.

Deep Reinforcement Learning Framework Our target is to develop a fast observation strategy, and we achieve it by learning a parameterized policy $\pi_{\theta}(a|s)$ optimizing the classification performance within a limited number of observa-

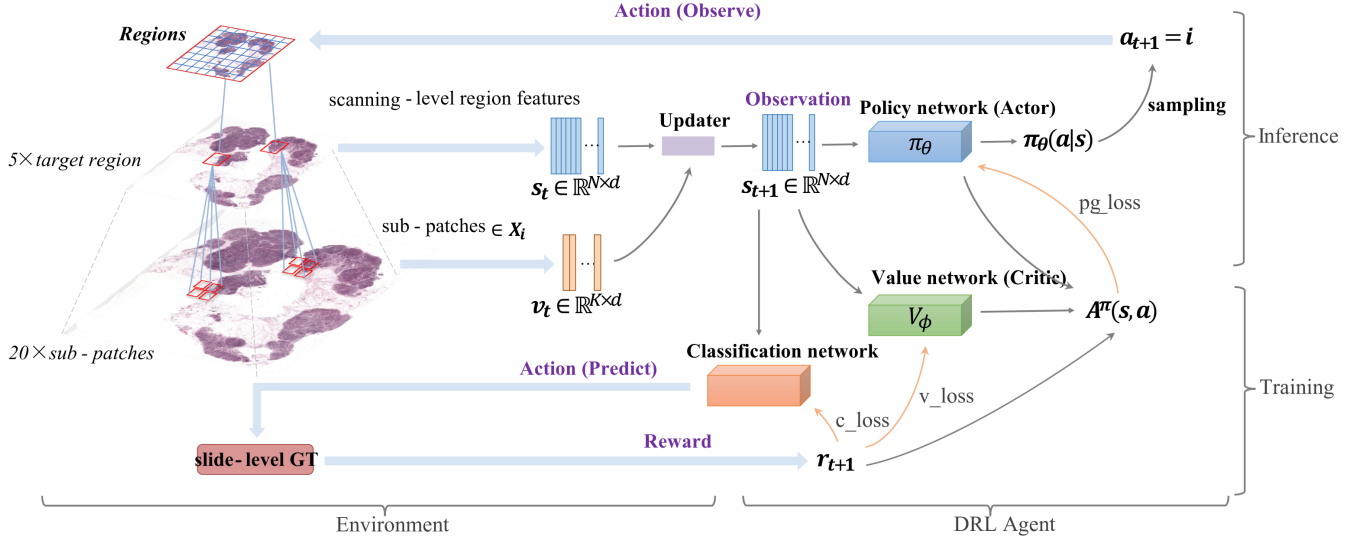


Figure 2: The architecture of RLogist. In the agent-environment loop, the blue arrows represent interaction with the environment, while the grey arrows represent the data pathway in the reinforcement learning model.

tion steps. The terminology $\pi_\theta(a|s)$ is defined by:

$$\pi_\theta(a|s) = P(A_t|S_t; \theta) \quad (1)$$

where A_t and S_t denotes the set of possible actions and states at time step t . Acting as the backbone of our proposed model, the stochastic policy network takes scanning-level feature embeddings s_t as the input at time step t , and outputs the logits of each legal actions, i.e., target regions for the next observation move. Note that, the explored regions need not be re-observed, we mask out them by replacing the logits of these unavailable actions with large negative number (e.g., -1×10^8) before sampling the action a_t . The process of an agent scanning the whole image at low-magnification is essentially to analyze multiple high-level regions in parallel with the policy network $\pi_\theta(a|s)$. After receiving the action a_t to observe a new target region, the state transits to the next state s_{t+1} by updating scanning-level feature with local feature updater f_{local} and unobserved region feature updater f_{global} , which will be described in detail within the following subsection. Through this environment design, we can handle the problem of changing action spaces that the number of low-magnification instances extracted from different WSIs after patching may be different. The agent-environment loop is summarized in Algorithm 1.

The learning task is to optimize the parameters θ that maximize the overall performance $J(\pi_\theta)$:

$$J(\pi_\theta) = \int_{\tau} P(\tau | \pi_\theta) R(\tau) = \mathbb{E}_{\tau \sim \pi_\theta} [R(\tau)] \quad (2)$$

where $R(\tau)$ is the expected return of the trajectory τ , associated with reward r_t . Given that only slide-level labels are available, we indirectly give the feedback (reward r_t) to the policy of observation path selection, using the classification results generated at each steps and the end of each trajectory.

Here we discuss the formation of rewards in detail and give corresponding proofs, as it is critical to design rewards

which are consistent with our ultimate target. Actually, the objective of finding a fast observation path can be modeled differently as these two optimization problems:

(1) Jointly optimize the slide-level prediction performance and the length of the observation path;

$$\theta = \arg \min_{\rho} [L_c, \ell] \quad (3)$$

(2) With a fixed length of the observation path, merely optimize the slide-level prediction performance

$$\theta = \arg \min_{\rho, \ell_0} L_c \quad (4)$$

where L_c is the classification loss, and ℓ is the length of the observation path. For binary classification tasks, the cross entropy loss can be calculated as:

$$L_c = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \quad (5)$$

For the former target, the agent may predict the slide-level label \hat{y}_t at each time step t , and an instant reward is given by comparing \hat{y}_t with the ground-truth y_t with cross entropy. This instant reward design can also be used on the latter objective to accelerate the convergence. However, if the model receives a direct reward after interacting with each selected regions, the agent will eventually learn a greedy strategy that may converge to the local optimum. Under certain circumstances, observing a specific region first may not be the best choice for prediction at this time step, but can help the agent to gain a better understanding of the whole slide that benefits following region-selecting decisions, resulting in better performance after certain steps than greedy strategies.

On common datasets, the trajectory length is relatively short for scanning at low-magnification (usually less than 100 time steps), making it easy to converge and feasible to just use a final reward at the done state when optimizing the policy. In the experiments, we only use the final reward of each trajectory to learning a more comprehensive strategy.

With this well-defined environment, it is crucial to adopt an effective while efficient policy gradient algorithm for the entire reinforcement learning process to optimally leverage the reward. We use the Proximal Policy Optimization (PPO) in our method due to its computational efficiency and satisfactory performance. Here, we briefly introduce how the PPO algorithm optimize the policy for region selection in the training stage. PPO-clip updates the policy π_θ via maximizing the objective $L(s, a, \theta_k, \theta)$:

$$\theta_{k+1} = \arg \max_{\theta} \mathbb{E}_{s, a \sim \pi_{\theta_k}} [L(s, a, \theta_k, \theta)] \quad (6)$$

Based on the idea of importance sampling (Kloek and Van Dijk 1978), the weight assigned to the current policy also depends on older policies. The probability ratio between previous and current policies is defined by:

$$\gamma(\theta, \theta_k) = \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} \quad (7)$$

Then $L(s, a, \theta_k, \theta)$ is given by:

$$L = \min \left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}(s, a)}, g(\epsilon, A^{\pi_{\theta_k}(s, a)}) \right) \quad (8)$$

where

$$g(\epsilon, A) = \begin{cases} (1 + \epsilon)A & A \geq 0 \\ (1 - \epsilon)A & A < 0 \end{cases} \quad (9)$$

ϵ is a relatively small hyperparameter to clip the objective, which roughly says how far away the new policy is allowed to go from the old.

Through optimizing the classification network with Eq. 5 and policy network with Eq. 8, respectively, our RLogist model can obtain the remarkable path-selection and classification performance only using the slide-level weak labels. The training pipeline of the DRL models (policy network π_θ and critic network V_ϕ) based on the actor-critic architecture and PPO algorithm is demonstrated in Algorithm 2. We compare the performance of PPO with classic policy gradient algorithms like REINFORCE (Luthans and Stajkovic 1999) in the experiments.

Fast Convergence with Pretrained Classifier At the end of an observation path, the slide-level label is predicted with the information of the observed regions by the classification network. However, training two networks from scratch simultaneously in an RL algorithm leads to convergence difficulties. During the cold start phase of model training, we apply a pretrained classifier to accelerate convergence. We train CLAM classifiers (Lu et al. 2021) on the training set of corresponding datasets in different magnifications. We split the classification multilayer perceptrons (MLPs) from the models and used them as the patch classifiers at different magnifications. By this means, the policy network is expected to focus on developing the path selection ability with a fixed classification network.

Unobserved Region Feature Update In our environment, the state transition process is to extract the high-power feature corresponding to the action, and then update the

Algorithm 2: DRL Training Pipeline

Input: Initial policy parameter θ_0 ; initial value function parameters ϕ_0 ;
Parameters: Starting state distribution ρ_0 ;
Output: The trained policy network;

- 1: **for** $k = 0, 1, 2, \dots$ **do**
 - 2: **for** $i = 0, 1, 2, \dots$ **do**
 - 3: Run policy $\pi_k = \pi(\theta_k)$ in the environment to generate an observation path $\tau_i \langle s, a, r, p, \rho_0 \rangle$.
 - 4: **end for**
 - 5: Collect a set of trajectories $D_k = \{\tau_i\}$.
 - 6: Estimate the advantage \hat{A}_t with current value function V_{ϕ_k} using Generalized Advantage Estimate.
 - 7: Update the policy by maximizing PPO Objective:
 $\theta_{k+1} = \arg \max_{\theta} \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^T L(s, a, \theta_k, \theta)$
 - 8: Compute Reward-to-go \hat{R}_t .
 - 9: Update the value function with MSE loss: $\phi_{k+1} = \arg \min_{\phi} \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^T (V_{\phi}(s_t) - \hat{R}_t)^2$
 - 10: **end for**
-

scanning-level representation with it. Firstly, the environment updates the scanning level feature of the currently-observed region, with the newly extracted high-power feature embedding of the sub-patches belonging to this region and the local feature updater f_{local} . f_{local} can be implemented as a Vision Transformer (Dosovitskiy et al. 2020) and pretrained by self-supervision, taking the sequence of feature embeddings of these sub-patches in the region a_t as the input:

$$\mathbf{v}'_{a_t} = f_{local}(\{\mathbf{v}_j^{a_t}\}_{j=1}^K) \quad (10)$$

where \mathbf{v}'_{a_t} is the updated local scanning-level feature embedding.

Then, with the pair of feature embedding $(\mathbf{v}_{a_t}, \mathbf{v}'_{a_t})$ for region a_t , the scanning-level representation of unobserved regions can be updated with f_{global} , implemented as an MLP-head. This MLP-head takes the concatenated vector $[\mathbf{v}_i, \mathbf{v}_{a_t}, \mathbf{v}'_{a_t}]$ as the input:

$$\hat{\mathbf{v}}'_i = f_{global}(\mathbf{v}_i, \mathbf{v}_{a_t}, \mathbf{v}'_{a_t}) \quad (11)$$

The objective of the MLP prediction is the local region feature directly generated from its corresponding high-magnification sub-patches. Therefore, the MLP-head can be pretrained with the ground-truth generated by f_{local} :

$$\mathbf{v}'_i = f_{local}(\{\mathbf{v}_j^i\}_{j=1}^K) \quad (12)$$

Experiments

In our experiments, we use two publicly available WSI datasets, i.e., the CAMELYON16 lymph node WSI dataset¹, and the TCGA-NSCLC lung cancer dataset², to demonstrate the effectiveness of our DRL fast observation strategy and the unobserved region feature update mechanism.

¹<https://camelyon16.grand-challenge.org/>

²<https://www.cancer.gov/tcga>

Dataset

CAMELYON16 is a typical dataset for breast-cancer lymph-node-metastasis detection. CAMELYON16 consists of 270 annotated whole slides for training and another 129 slides as a held-out official test set. We directly use the officially divided training set and test set in our experiments. After segmentation and patching, the dataset yields roughly 0.25 million patches at 5 \times magnification and 3.2 million patches at 20 \times magnification with on average about 8,000 and 625 patches per WSI.

The TCGA-NSCLC dataset consists of 1054 diagnostic WSIs from the TCGA NSCLC repository under the TCGA-LUSC and TCGA-LUAD projects. 10 low-quality slides without MPP and magnification information are discarded. We randomly split the dataset into 835 training slides and 209 testing slides with stratified sampling from 2 different TCGA projects. After pre-processing, a total of about 0.3 million patches at 5 \times magnification and 4.8 million patches at 20 \times magnification with on average about 350 and 5600 patches per WSI were obtained.

For both the datasets mentioned above, every WSI is cropped into 224 \times 224 patches without overlap and only slide-level labels are used to train our models.

Experimental Setup

We adopt the ImageNet pretrained ResNet-50 (He et al. 2016) for the feature extraction network as f , which converts patches into 1024-dimensional features v . For DRL, we implement policy gradient algorithm PPO2 roughly matching the details in stable-baselines3 (Raffin et al. 2021). We also refer to the code-level optimizations presented in openai/baselines’s PPO (Dhariwal et al. 2017), implementation details in CleanRL (Huang et al. 2021) and some of the tricks summarized by Huang et al. (2022). We use the Adam optimizer with an annealing learning rate, which is initialized as 2.5×10^{-4} and decays to 0 as the number of time steps increases, and $\epsilon = 1 \times 10^{-5}$ to update the model weights during the training of our policy network. Cross-entropy loss is adopted to calculate the trajectory reward. We refer to the API design of OpenAI Gym (Brockman et al. 2016) to implement our RL environment. For CLAM classifier, we follow the implementation details and experiment settings of Lu et al. (2021). At the training stage, the CLAM pretrained classifier at low magnification (scanning level) is adopted to help initialize the policy network. All models in the experiment are implemented in PyTorch and trained until convergence on half NVIDIA Tesla T4 GPU.

Slide-Level Classification Results

We evaluate and compare our method to a collection of baselines including (1) standard MIL algorithm using traditional aggregators such as mean-pooling or max-pooling and (2) recent state-of-the-art deep MIL models, on the tasks of WSI classification. We have trained 3 DRL agents that observe different proportions 10%, 20%, 50% of the regions in each WSI, i.e. with fixed length of the observation paths, before making a final slide-level prediction, respectively.

Method	CAMELYON16		TCGA-NSCLC	
	Acc.	AUC	Acc.	AUC
Mean-pooling	0.6389	0.4647	0.7282	0.8401
Max-pooling	0.8062	0.8569	0.8593	0.9463
ABMIL	0.8682	0.8760	0.7719	0.8656
DSMIL	0.7985	0.8179	0.8058	0.8925
TransMIL	0.8449	0.8769	0.8565	0.9303
CLAM-SB	0.8760	0.8809	0.8180	0.8818
RLogist-0.1	0.8242	0.8287	0.8284	0.8920
RLogist-0.2	0.8624	0.8794	0.8389	0.9026
RLogist-0.5	0.8685	0.8874	0.8405	0.9031

Table 1: Classification results on CAMELYON16 and TCGA-NSCLC. The RLogist-0.1, RLogist-0.2, and RLogist-0.5 denote the models trained to observe different time steps before making a final prediction, corresponding to different proportions of the regions in WSI. For fairness, we do not compare our method with other methods that learn the high-level cross-instance information fusion, which may achieve better classification performance. ^a

^aAs one of the reviewers suggested, we report the results of TransMIL based on our implementation, and the results are slightly different from the originally reported results.

Method	Average Runtime / min
Mean/Max-pooling	39.91
CLAM-SB	40.89
RLogist-0.2	9.74

Table 2: Average inference time per WSI for slide-level classification on the test set, using half NVIDIA Tesla T4 GPU. Here "half" indicates the half computing power provided by a virtual GPU, which consumes much more time on feature extraction, compared to using a more powerful GPU, e.g., Tesla V100/A100.

As shown in Table 1, the performance of our RLogist is comparable to state-of-the-art methods (ABMIL (Ilse, Tomczak, and Welling 2018), DSMIL (Li, Li, and Eliceiri 2021), TransMIL (Shao et al. 2021), and CLAM-SB (Lu et al. 2021)), while having a significantly shortened observation path. RLogist-0.1 represents the agent only observes 10% of the regions in the WSI before generating slide-level classification results (0.2 and 0.5, respectively).

We illustrate our fast observation strategy by comparing the run time of different methods in the inference stage. Experiments are performed on the CAMELYON16 test set, using the ImageNet pretrained ResNet-50 to extract 1024-dimensional features. The average inference run time for each WSI is reported in Table 2. Note that the run time here includes the time consumption of WSI data preprocessing (segmentation and patching), feature extraction, and classification. As is shown, our RLogist reduces the burden of feature extraction significantly with a short observation path.

As is shown in Figure 3, the RLogist has observed most of the diagnostic-relevant (positive) regions in the first 10

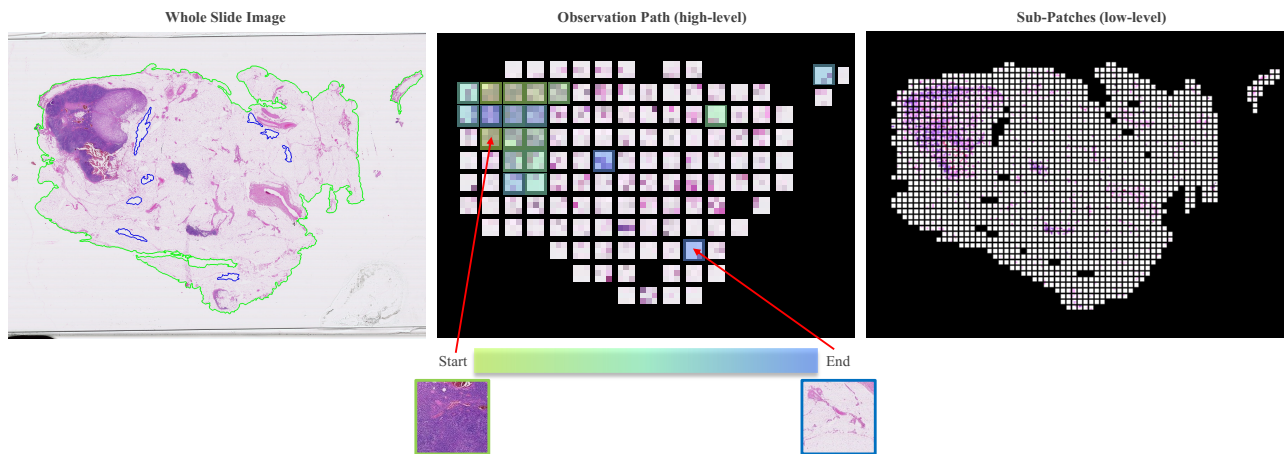


Figure 3: Visualized observation path of the RLogist on lymph node metathesis WSIs. The RLogist-0.2 starts the observation path from the region marked in green and ends in the blue region. About 20% of the patches were covered to make slide-level label predictions. It also seems that less than 20 steps can complete diagnosis for this WSI.

Observation Strategy	AUC	Convergence Episodes
Random (baseline)	0.7870	/
REINFORCE	0.8364	142k
PPO	0.8794	25k

Table 3: Ablations with different observation strategies.

steps, which are marked in green. For the vast majority of metastatic cases, positive areas are detected within 20 steps.

Ablation Study

To further determine the contributions of the DRL observation strategy and the feature update mechanism, we have conducted a series of ablation studies. All ablation study experiments are based on the CAMELYON16 dataset, for the positive slides in this dataset are highly unbalanced (a large portion of negative patches in a positive WSI) and the effect of fast observation strategy may be reflected better.

Observation Strategy with DRL The observation policy is trained to correctly select regions that are either directly diagnostic-relevant or helpful for subsequent observation moves. Here, we compared the effect of a random observation strategy and the policies trained with our proposed method using REINFORCE and PPO algorithm. For the random strategy, we randomly sample a proportion of regions from the WSI (consistent with the time steps of the RL agent), and predict the slide-level labels with CLAM using high-magnification feature embeddings of the sub-patches in these regions. The agent is trained to observe on average 20% of the WSI foreground (RLogist-0.2). The classification and convergence results are shown in Table 3.

Compared with the random strategy, it can be seen that both REINFORCE and PPO algorithm can improve the classification performance, and PPO is more effective in diagnosis prediction and much easier to converge.

Feature Update Mechanism Here we explore actual improvements for feature updaters f_{local} and f_{global} . The per-

Feature Updater	AUC
Fixed feature (baseline)	0.8514
f_{local}	0.8559
$f_{local} + f_{global}$	0.8794

Table 4: Ablations with different feature updaters.

formance of the model under different configurations is shown in Table 4. Compared with the model without feature updater, it can be seen that both feature update mechanisms can enhance the classification performance, up to 0.53% for f_{local} and 2.74% for $f_{local} + f_{global}$.

Conclusions and Future Work

We present a deep reinforcement learning method, named RLogist, for whole-slide image analysis. RLogist simulates a pathologist’s decision-making process and can quickly complete the reading path navigation through the fusion of cross-magnification image information. We design and implement the environment for RL training with discretized action space and conditional feature super-resolution mechanism. Competitive classification results have been achieved on two widely-used benchmark datasets, demonstrating the effectiveness and feasibility of our method. Notably, our method can significantly reduce the observation scale of image patches without losing much classification performance. Although recent AI-based screening/diagnostic algorithms have achieved satisfactory results on benchmark datasets, these data-driven methods, including ours, should be used with caution before being trained with large-scale clean data and extensive evaluated with multi-center clinical data.

In the future, we would like to construct a better RL environment with fused cross-instance observation space using Transformers (Vaswani et al. 2017), so as to enhance the feature representation. Moreover, it would be more reasonable to plan the observation path with variable length for each WSI as indicated in Eq. 3 and Fig. 3. More sophisticated RL tricks should be used to avoid learning the wrong shortcuts.

References

- Abdi, A. H.; Saha, P.; Srungarapu, V. P.; and Fels, S. 2020. Muscle excitation estimation in biomechanical simulation using NAF reinforcement learning. In *Computational Biomechanics for Medicine*, 133–141. Springer.
- Alansary, A.; Oktay, O.; Li, Y.; Le Folgoc, L.; Hou, B.; Vailant, G.; Kamnitsas, K.; Vlontzos, A.; Glocker, B.; Kainz, B.; et al. 2019. Evaluating reinforcement learning agents for anatomical landmark detection. *Medical image analysis*, 53: 156–164.
- Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.
- Bulten, W.; Kartasalo, K.; Chen, P.-H. C.; Ström, P.; Pinckaers, H.; Nagpal, K.; Cai, Y.; Steiner, D. F.; van Boven, H.; Vink, R.; et al. 2022. Artificial intelligence for diagnosis and Gleason grading of prostate cancer: the PANDA challenge. *Nature medicine*, 28(1): 154–163.
- Campanella, G.; Hanna, M. G.; Geneslaw, L.; Mirafior, A.; Werneck Krauss Silva, V.; Busam, K. J.; Brogi, E.; Reuter, V. E.; Klimstra, D. S.; and Fuchs, T. J. 2019. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8): 1301–1309.
- Carbonneau, M.-A.; Cheplygina, V.; Granger, E.; and Gagnon, G. 2018. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77: 329–353.
- Chen, R. J.; Chen, C.; Li, Y.; Chen, T. Y.; Trister, A. D.; Krishnan, R. G.; and Mahmood, F. 2022. Scaling Vision Transformers to Gigapixel Images via Hierarchical Self-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16144–16155.
- Chen, Z.; Zhang, J.; Che, S.; Huang, J.; Han, X.; and Yuan, Y. 2021. Diagnose like a pathologist: Weakly-supervised pathologist-tree network for slide-level immunohistochemical scoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 47–54.
- Cheng, K.; Iriondo, C.; Calivá, F.; Krogue, J.; Majumdar, S.; and Pedoia, V. 2019. Adversarial policy gradient for deep learning image augmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 450–458. Springer.
- Coronato, A.; Naeem, M.; De Pietro, G.; and Paragliola, G. 2020. Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109: 101964.
- Dhariwal, P.; Hesse, C.; Klimov, O.; Nichol, A.; Plappert, M.; Radford, A.; Schulman, J.; Sidor, S.; Wu, Y.; and Zhokhov, P. 2017. OpenAI Baselines. <https://github.com/openai/baselines>.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissensborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Feng, J.; and Zhou, Z.-H. 2017. Deep MIML network. In *Thirty-First AAAI conference on artificial intelligence*.
- Furuta, R.; Inoue, N.; and Yamasaki, T. 2019. Pixelrl: Fully convolutional network with reinforcement learning for image processing. *IEEE Transactions on Multimedia*, 22(7): 1704–1719.
- Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.
- Hashimoto, N.; Fukushima, D.; Koga, R.; Takagi, Y.; Ko, K.; Kohno, K.; Nakaguro, M.; Nakamura, S.; Hontani, H.; and Takeuchi, I. 2020. Multi-scale domain-adversarial multiple-instance CNN for cancer subtype classification with unannotated histopathological images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3852–3861.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Huang, S.; Dossa, R. F. J.; Raffin, A.; Kanervisto, A.; and Wang, W. 2022. The 37 Implementation Details of Proximal Policy Optimization. In *ICLR Blog Track*. <https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/>.
- Huang, S.; Dossa, R. F. J.; Ye, C.; and Braga, J. 2021. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *arXiv preprint arXiv:2111.08819*.
- Ilse, M.; Tomczak, J.; and Welling, M. 2018. Attention-based deep multiple instance learning. In *International conference on machine learning*, 2127–2136. PMLR.
- Joos, E.; Péan, F.; and Goksel, O. 2020. Reinforcement learning of musculoskeletal control from functional simulations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 135–145. Springer.
- Kloek, T.; and Van Dijk, H. K. 1978. Bayesian estimates of equation system parameters: an application of integration by Monte Carlo. *Econometrica: Journal of the Econometric Society*, 1–19.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Li, B.; Li, Y.; and Eliceiri, K. W. 2021. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14318–14328.
- Li, Y. 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.
- Lin, Z.; Li, J.; Shi, J.; Ye, D.; Fu, Q.; and Yang, W. 2022. JueWu-MC: Playing Minecraft with Sample-efficient Hierarchical Reinforcement Learning. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference*

- on *Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 3257–3263. ijcai.org.
- Liu, D.; and Jiang, T. 2018. Deep reinforcement learning for surgical gesture segmentation and classification. In *International conference on medical image computing and computer-assisted intervention*, 247–255. Springer.
- Lu, M. Y.; Williamson, D. F.; Chen, T. Y.; Chen, R. J.; Barbieri, M.; and Mahmood, F. 2021. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6): 555–570.
- Luthans, F.; and Stajkovic, A. D. 1999. Reinforce for performance: The need to go beyond pay and even rewards. *Academy of Management Perspectives*, 13(2): 49–57.
- Pineda, L.; Basu, S.; Romero, A.; Calandra, R.; and Drozdal, M. 2020. Active MR k-space sampling with reinforcement learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 23–33. Springer.
- Pinheiro, P. O.; and Collobert, R. 2015. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1713–1721.
- Qaiser, T.; and Rajpoot, N. M. 2019. Learning where to see: A novel attention model for automated immunohistochemical scoring. *IEEE transactions on medical imaging*, 38(11): 2620–2631.
- Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268): 1–8.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Shao, Z.; Bian, H.; Chen, Y.; Wang, Y.; Zhang, J.; Ji, X.; et al. 2021. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in Neural Information Processing Systems*, 34: 2136–2147.
- Shen, C.; Gonzalez, Y.; Chen, L.; Jiang, S. B.; and Jia, X. 2018. Intelligent parameter tuning in optimization-based iterative CT reconstruction via deep reinforcement learning. *IEEE transactions on medical imaging*, 37(6): 1430–1439.
- Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676): 354–359.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, J.; Miao, J.; Yang, X.; Li, R.; Zhou, G.; Huang, Y.; Lin, Z.; Xue, W.; Jia, X.; Zhou, J.; et al. 2020. Auto-weighting for breast cancer classification in multimodal ultrasound. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 190–199. Springer.
- Wang, X.; Yang, S.; Zhang, J.; Wang, M.; Zhang, J.; Huang, J.; Yang, W.; and Han, X. 2021. Transpath: Transformer-based self-supervised learning for histopathological image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 186–195. Springer.
- Xie, C.; Muhammad, H.; Vanderbilt, C. M.; Caso, R.; Yarlaga, D. V. K.; Campanella, G.; and Fuchs, T. J. 2020. Beyond classification: Whole slide tissue histopathology analysis by end-to-end part learning. In *Medical Imaging with Deep Learning*, 843–856. PMLR.
- Yang, H.; Shan, C.; Kolen, A. F.; et al. 2020. Deep q-network-driven catheter segmentation in 3d us by hybrid constrained semi-supervised learning and dual-unet. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 646–655. Springer.
- Ye, D.; Chen, G.; Zhang, W.; Chen, S.; Yuan, B.; Liu, B.; Chen, J.; Liu, Z.; Qiu, F.; Yu, H.; et al. 2020a. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 621–632.
- Ye, D.; Liu, Z.; Sun, M.; Shi, B.; Zhao, P.; Wu, H.; Yu, H.; Yang, S.; Wu, X.; Guo, Q.; et al. 2020b. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6672–6679.
- Ye, J.; Xue, Y.; Long, L. R.; Antani, S.; Xue, Z.; Cheng, K. C.; and Huang, X. 2020c. Synthetic sample selection via reinforcement learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 53–63. Springer.
- Zaech, J.-N.; Gao, C.; Bier, B.; Taylor, R.; Maier, A.; Navab, N.; and Unberath, M. 2020. Learning to Avoid Poor Images. In *Bildverarbeitung für die Medizin 2020*, 185–185. Springer.
- Zhao, Y.; Yang, F.; Fang, Y.; Liu, H.; Zhou, N.; Zhang, J.; Sun, J.; Yang, S.; Menze, B.; Fan, X.; et al. 2020. Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4837–4846.
- Zhao, Z.; Gan, C.; Wu, J.; Guo, X.; and Tenenbaum, J. B. 2021. Augmenting policy learning with routines discovered from a single demonstration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11024–11032.
- Zhou, S. K.; Le, H. N.; Luu, K.; Nguyen, H. V.; and Ayache, N. 2021. Deep reinforcement learning in medical imaging: A literature review. *Medical image analysis*, 73: 102193.
- Zhu, F.; Guo, J.; Xu, Z.; Liao, P.; Yang, L.; and Huang, J. 2018. Group-driven reinforcement learning for personalized mhealth intervention. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 590–598. Springer.