Interpretable Inference and Classification of Tissue Types in Histological Colorectal Cancer Slides Based on Ensembles Adaptive Boosting Prototype Tree

Meiyan Liang, Ru Wang, Jianan Liang, Lin Wang, Bo Li, Xiaojun Jia, Yu Zhang, Qinghui Chen, Tianyi Zhang, and Cunlin Zhang

***Abstract*—Digital pathology images are treated as the "gold standard" for the diagnosis of colorectal lesions, especially colon cancer. Real-time, objective and accurate inspection results will assist clinicians to choose symptomatic treatment in a timely manner, which is of great significance in clinical medicine. However, Manual methods suffers from long inspection cycle and serious reliance on subjective interpretation. It is also a challenging task for existing computer-aided diagnosis methods to obtain models that are both accurate and interpretable. Models that exhibit high accuracy are always more complex and opaque, while interpretable models may lack the necessary accuracy. Therefore, the framework of ensemble adaptive boosting prototype tree is proposed to predict the colorectal pathology images and provide interpretable inference by visualizing the decision-making process in each base learner. The results showed that the proposed method could effectively address the "accuracy-interpretability trade-off" issue by ensemble of *m* adaptive boosting neural prototype trees. The superior performance of the framework provides a novel paradigm for interpretable inference and high-precision prediction of pathology image patches in computational pathology.**

***Index Terms*—Adaptive boosting, colorectal cancer, ensemble learning, interpretable inference, prototype tree.**

Manuscript received 22 September 2022; revised 1 April 2023 and 5 June 2023; accepted 17 October 2023. Date of publication 23 October 2023; date of current version 6 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 11804209, in part by the Natural Science Foundation of Shanxi Province under Grants 201901D211173, 201901D211172, and 202103021223411, and in part by the Research Project Supported by Shanxi Scholarship Council of China under Grant 2023-010. *(Corresponding authors: Meiyan Liang; Lin Wang; Xiaojun Jia.)*

Meiyan Liang, Ru Wang, Jianan Liang, Xiaojun Jia, Yu Zhang, Qinghui Chen, and Tianyi Zhang are with the School of Physics and Electronic Engineering, Shanxi University, Taiyuan 030006, China (e-mail: meiyanliang@sxu.edu.cn; w425902505@163.com; liangjianan@sxu.edu.cn; jiaxj@sxu.edu.cn; 1827575063@qq.com; c13253754272@163.com; tianyizhangsxu@163.com).

Lin Wang is with the Shanxi Bethune Hospital, Shanxi Academy of Medical Sciences, Tongji Shanxi Hospital, Third Hospital of Shanxi Medical University, Taiyuan 030032, China (e-mail: 13613411069@139.com).

Bo Li is with the Department of Rehabilitation Treatment, Shanxi Rongjun Hospital, Taiyuan 030000, China (e-mail: mvplibo@163.com).

Cunlin Zhang is with the Beijing Key Laboratory for Terahertz Spectroscopy and Imaging, Key Laboratory of Terahertz, Optoelectronics, Ministry of Education, Capital Normal University, Beijing 100048, China (e-mail: cunlin_zhang@cnu.edu.cn).

Digital Object Identifier 10.1109/JBHI.2023.3326467

I. Introduction

ACCORDING to the "GLOBOCAN 2020" released by International Agency for Research on Cancer (IARC) of World Health Organization (WHO), it is estimated 1.9 million new colorectal cancer (including anus) cases and 935000 deaths were estimated to occur in 2020, representing about one in ten cancer cases and deaths. Overall, colorectal cancer ranks third in terms of incidence, but second in terms of mortality worldwide [1]. However, the clinical prediction of colorectal cancer mainly relies on manual feature extraction and analyzation of pathological images, suffering from long inspection cycle and severe reliance on subjective interpretation, which cannot meet the requirements of precision medicine. Therefore, it is of great significance to establish a real-time, objective, and accurate pathological image prediction model that provides human-interpretable features automatically in medical image analysis [2], [3], [4]. Deep convolutional networks (ConvNets) [5] have been widely applied in large-scale image classification tasks [6], [7], [8], [9] due to their excellent prediction performance and fewer parameters [10], [11]. In ConvNets, the neurons between the convolutional layers are connected by a weight sharing mechanism and activated by nonlinear functions. Therefore, the input image is progressively transformed into semantic features through multiple layers, resulting in its lack of interpretability. Hence, deep convolutional neural networks are generally perceived as a "black boxes" that can approximate any nonlinear function. A neural network capable of providing interpretability for learned functions is an important criterion for evaluating the reliability of the neural networks [12], [13], [14]. Therefore, it is of great significance to carry out the interpretability studies of deep learning in practical applications, especially in medical image processing.

In computer vision tasks, various interpretability strategies have been proposed for different notions of interpretability [15], [16], [17], [18]. In 2009, Erhan [19] et al. presented an activation maximization to visualize the features learned by the deep learning model, which utilized the gradient ascent algorithm to generate feature representations with higher activation values of neurons in each layer. It can also be applied to visualize the

learned features of the unsupervised deep learning models, such as Deep Belief Network (DBN) [20]. In 2014, Zeiler [21] et al. introduced a deconvolutional network that maps the activity of the convolutional layers in a ConvNet back to the input pixel space, showing which input patterns originally led to a given activation in the feature map. Afterwards, Simonyan [22] et al. extended the application of deconvolution-based visualization approaches to obtain their relationship with the gradient-based visualization methods [23], [24], and concluded that gradient based visualization can be regarded as the generalization of deconvolution based methods [25]. This is because the gradient based visualization techniques can be applied to any active layer, not just the convolutional layer. Other gradient based methods include a series of algorithms related to class activation maps (CAMs) [26] to visualize the learned pattern of model. In 2019, Selvaraju [27] et al. developed a Gradient Weighted Class Activation Mapping (Grad-CAM) to visualize the learned features of the attention-based model. The heat map uses the gradient of the target concept following the final convolutional layer to generate a coarse localization map, highlighting important regions of the image to predict the target concept. However, the interpretable models either based on activation maximization or gradient explain input images by visualizing the learned features, which can only be applied to a trained network and cannot provide the inference process of the model. Whereas, the attention-based deep learning model can only visualize the key parts of the input image, thus lacking interpretability for the overall category.

To improve the reliability of the model, recent interpretable inference study is mainly based on the intrinsic understanding of the learned patterns, such as concept attributions and prototypes. In 2017, Bau [28], [29] et al. attempted to quantify the interpretability of latent representations in CNNs by measuring the overlap between highly activated image regions and labeled visual concepts. However, this method required fine-grained manual labeling for large datasets specific to network purposes. In 2018, Li et al. [30] proposed an interpretable autoencoder architecture with an embedded prototype layer to generate feature representations for learned prototype in latent space. Afterwards, the learned prototype can be visualized by inverse transform of the feature representations using decoder. Since this approach of generating interpretable prototypes with decoder is derived from the features in the latent space, it cannot map the prototypes to the practical training image space, resulting in a lack of human-interpretability for the generated prototypes. Therefore, Chen et al. [31] introduced a Prototype Part Network (ProtoPNet) architecture, which searched for image parts in the training set corresponding to learned prototypes in the latent space to define the interpretability. However, these post-hoc interpretable methods can only visualize the learned features of the "black box" for final prediction. The causes of learned features cannot be tracked if the model produces interpretations that do not make sense to human experts. Meanwhile, deep learning models that exhibit high accuracy are always more complex and opaque, while interpretable models may lack the necessary accuracy. It is a challenging task for existing computer-aided diagnosis methods to obtain models that are both accurate and interpretable. To address this issue, an ensembles of adaptive boosting prototype tree network is proposed to perform fine-grained identification for multi-class image patches in colorectal pathology slides. The ensemble based method can not only achieve high-precision prediction, but also provide intrinsic interpretability for the prediction results through the visualization of decision-making process. Meanwhile, it does not require manual labeling and is very similar to human reasoning process, which opens up new perspectives for reliable inferential diagnostic methods. Our scientific contributions are as follows:

1) An Ensemble Adaptive Boosting neural prototype tree architecture is proposed for fine-grained pathological image classification, which combines neural prototype tree with adaptive boosting algorithm, to improve the overall prediction accuracy of the model and reduce bias through sequentially focusing on poor predictions in the previous base learner and attempting to rectify them in the following round.
2) The inference process of the ensemble model is visualized by prototypical parts from *m* subtrees based on the shifted data distribution. The interpretability of the ensemble model can also be improved by the prototypes obtained from the complementary base learners. Meanwhile, this approach also provides the class-related discriminative features for the classification results.
3) We address the "accuracy-interpretability trade-off" issue by ensemble of *m* adaptive boosting neural prototype trees [31]. The method can not only effectively improve the prediction accuracy for pathological image patches by adaptive boosting algorithm, but also significantly enhance the reliability by visualizing the intrinsically interpretable reasoning process the base learners. The superior performance of the framework opens a new perspective for high-precise and reliable inferential diagnosis.

II. Principle of Adaptive Boosting Ensemble Learning

Boosting was proposed in the computational learning theory literature [32], [33], [34], which sequentially applying a typical classification algorithm to reweighted training data distribution and then adopting a weighted voting strategy for the sequence classifiers. This strategy can combine the performance of many "weak" classifiers to produce a strong "committee" through an additive model. Therefore, the adaptive boosting ensemble learning can reduce the bias compared to using only a single base learner.

Given a training dataset $D = [x, y] = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the number of training data, x_i and y_i are the *i*th input image and corresponding class label. Where $y_i \in \{0, 1, \ldots, k, \ldots, K-1\}$. If the number of classes $K > 2$, it is known as a multi-class classification problem. Here, the multinomial logit model has been applied for multi-class pathology image classification issue. The class probabilities that the *i*-th sample is predicted to be the *k*-th class can be given by:

$$p_{i,k} = \Pr(y_i = k|x_i) = \frac{e^{F_{i,k}(x_i)}}{\sum_{k=0}^{K-1} e^{F_{i,k}(x_i)}} \tag{1}$$

and then predict each class label according to

$$\hat{y}_i = \arg\max_k p_{i,k} \tag{2}$$

Here, $\hat{y}_i$ is the predicted label of input image x_i. The classification error occurs if $\hat{y}_i \neq y_i$. In (1), $F_{i,k} = F_{i,k}(x_i)$ denote

the learned function from the training data. $\Pr(y_i = k|x_i)$ indicates the probability that sample x_i is predicted as the kth class. For traditional logistic regression, it simply assume that $F_{i,k}(x_i) = \omega_k^T x_i$ (ω is the learned parameter). Therefore, the subscript of $F_{i,k}$ indicates that the function is learned from the training set $\{x_i\}_{i=1}^N$, which has a total of k categories. For ensemble learning framework, it can be expressed as $F^{(m)}(x)$ when it is an ensemble representation of m base learners.

Boosting algorithm builds an additive model of the "weak" classifiers (base learners) in a forward stage-wise fashion. It improves the overall performance of the model by iteratively combining "weak" classifiers to form a strong learner in a weighted manner. Therefore, the adaptive boosting model with m base learners can be written as the sum of m terms:

$$F^{(m)}(x) = \sum_{m=1}^{m} \beta_m b_m(x;\gamma_m) = F^{(m-1)}(x) + \beta_m b_m(x;\gamma_m) \tag{3}$$

Where $F^{(m-1)}(x)$ is the ensemble of the previous m-1 base learners. $b_m(x;\gamma_m)$ denotes the mth base learner, typically a regression tree for inference based on interpretable features. Here, γ_m and β_m are a set of learned parameters and weight of the mth weak learner, respectively. These parameters are all learned iteratively through the forward stage-wise distribution. m is the total number of base learners.

Hence, learning the additive regression model is to minimize the loss function of the following at each stage m:

$$\{\beta_m^*, \gamma_m^*\} = \arg\min_{\beta_m,\gamma_m} \sum_{i=1}^{N} L_{class}(y_i, F^{(m-1)} + \beta_m b_m(x_i;\gamma_m)) \tag{4}$$

Where β_m^* and γ_m^* are the optimized weights and learned parameters of mth base learner in ensemble model, respectively. Here, parameters $\{\beta_m^*\}_{m=1}^m$ and $\{\gamma_m^*\}_{m=1}^m$ can be obtained recursively by the forward stage-wise training process according (4). It is equivalent to minimizing the *negative multinomial log-likelihood* (*NML*) *loss:*

$$L_{class} = -\sum_{i=1}^{N} r_{i,k} \log p_{i,k} \tag{5}$$

Where $r_{i,k}$ is a one-bit binary variable indicating whether the model has made correct predictions. Therefore, $r_{i,k}$= 1 if $y_i = k$ and $r_{i,k}$= 0 when $y_i \neq k$.

Finally, the optimized $\{\beta_m^*\}_{m=1}^m$ and $\{\gamma_m^*\}_{m=1}^m$ are plugged in (3) for ensemble prediction.

III. Model

A. The Framework of Ensembles Adaptive Boosting Prototype Tree

Pathology slides diagnosis is based on multiple microscopic structures and macroscopic features of tissues, such as the morphology of cell nuclei, the ratio of nucleus to cytoplasm, the geometry of gland and texture of tissue. As pathological images from different organs have specific features, and the patterns in colon histopathological slides are more complex than others. Therefore, building a multi-classification model for colorectal pathological tissue images and visualizing interpretable inference process are of great significance in clinical applications. Here, an Ensembles Adaptive Boosting Prototype Tree (EnABPT) framework is proposed based on adaptive boosting algorithm to achieve high-precision prediction for multi-class colon image patches in digital pathology slides. Meanwhile, the model also uses interpretable features as splitting criteria of prototype tree to visualize the decision-making process, thereby obtaining the entire inference process of the model.

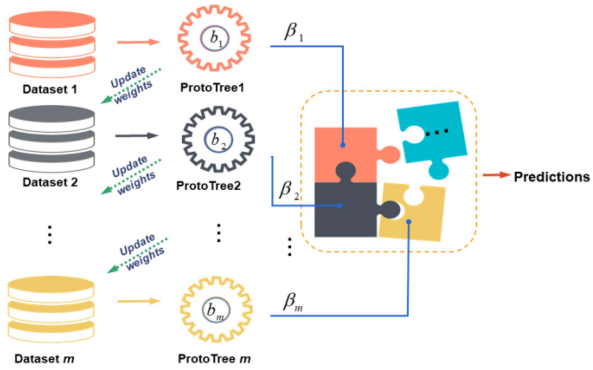

Fig. 1. Block diagram of ensembles adaptive boosting prototype tree.

The block diagram of Ensembles Adaptive Boosting Prototype Tree is shown in Fig. 1. The framework contains m Prototype Tree as base learners. During the iterative process, the base learners sequentially applies the backpropagation algorithm to the reweighted training data and then obtains the final prediction results by a weighted soft voting strategy for these sequence classifiers. The parameters γ_m and β_m in each base learner are also automatically obtained based on a forward-stage additive regression model using a negative log-likelihood loss between the predicted labels $\{\hat{y}_i\}_{i=1}^N$ and the ground truth $\{y_i\}_{i=1}^N$. In our case, γ_m is the learning parameter of mth base learner, which includes the splitting variables, split points, the constants in each leaf node and number of leaf nodes for each prototype tree. The parameter β_m is the learned weight of the base learner. In our case, not only the base classifiers but also the training samples are weighted in each base learner. Therefore, reweighted training data distribution means that the probability of sampling the confused samples will be adaptively boosted by the weighted resampling strategy. Here, instead of passing sample weights to the base learner, the training data can be resampled to reflect the sample weight. The training dataset for each base learner is the same size as the original dataset, which is created by sampling and replacing the original training dataset. The probability of each example being selected is proportional to its assigned weight. Specifically, the process is as follows:

i) In the 1st base learner, we assign each example an identical weight, which is set to 1/N. Here, N denotes number of examples in the training set. This means that the importance of correctly classifying samples is not emphasized in the 1st round.

ii) After training process, the amount of say for this base learner is calculated using the following formula:

$$\beta = 0.5 \cdot \log((1 - Er)/Er) \tag{6}$$

Here, Er and β denote the classification error and the amount of say (it is also known as coefficient/weight) of the base learner, respectively. Where the amount of say depends on how well the base learner classifies examples.

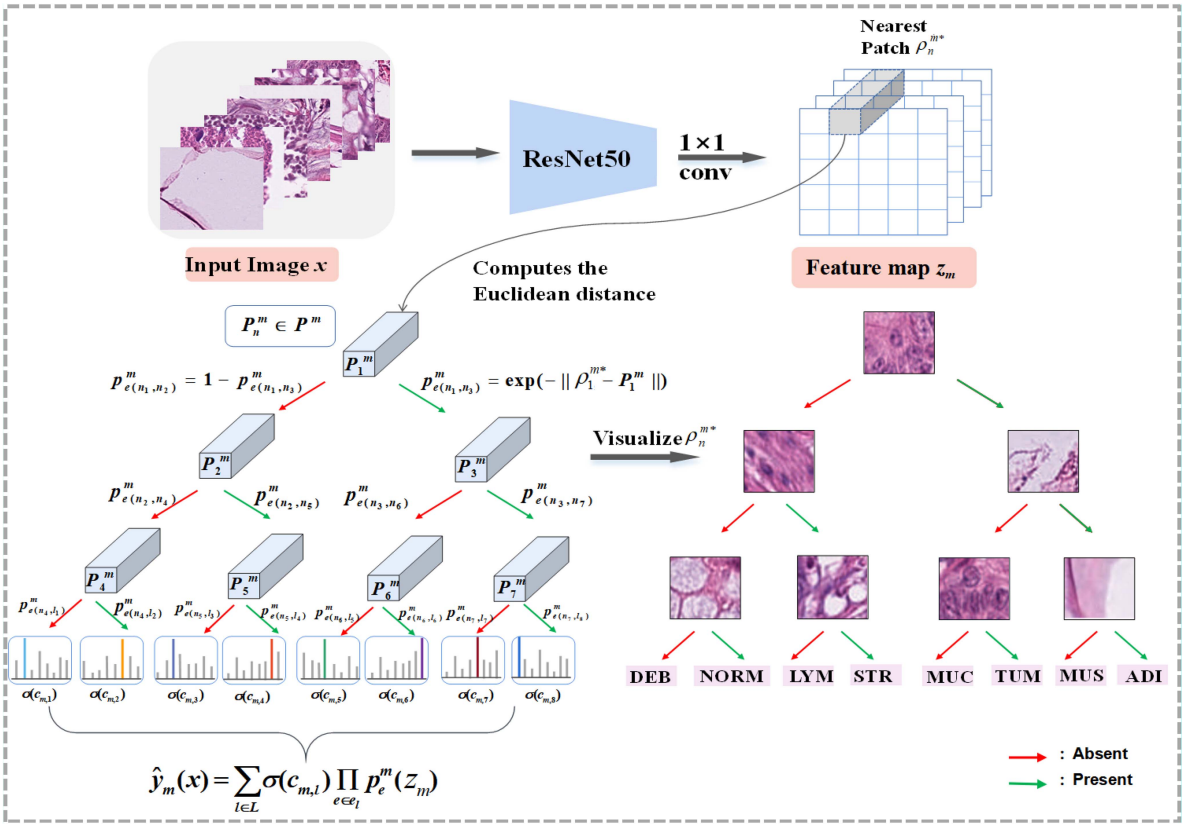

Fig. 2. Interpretable inference process of the prototype tree.

This formula can be generalized to the m-th base learner, thus, the weight of the m-th base learner is given by:

$$\beta_m = 0.5 \cdot \log\left((1 - Er_m)/Er_m\right) \tag{7}$$

Here, it is obviously that β_m decreases as Er_m increases. That is, weak learners with larger classification errors are assigned lower weights in the ensemble model.

iii) In the $(m+1)$-th round, for the correctly classified example in previous round, the weights are updated as:

$$UpdatedWeight_{m+1} = SampleWeight_m \cdot \exp(-\beta_m) \tag{8}$$

While for examples that were incorrectly classified in previous round, the sample weight is updated using the formula:

$$UpdatedWeight_{m+1} = SampleWeight_m \cdot \exp(\beta_m) \tag{9}$$

Note that the updated weights for correctly classified examples are lower than that of incorrectly classified examples. This means that the following base learner focuses more on incorrectly classified examples in the previous round. Therefore, a new dataset is created based on the previous one such that it contains more examples that were misclassified by the previous base learner. Then, we normalize the updated sample weights such that they sum to 1.

iv) A random number is sampled in the interval [0,1]. Then, we obtain which weight range the number falls within when the sample weights are viewed as a distribution. Here, the probability of the sample being sampled is proportional to the assigned weight of that sample. Therefore, the probability of misclassified samples in the previous round will be significantly increased in the following round.

The above process is repeated until the input datasets for each base learner are obtained. It is obviously that the data distribution have been shifted through the weighted resampling strategy.

B. The Principle of Neural Prototype Tree

In this framework, each base learner $b_m(x;\gamma_m)$ is a neural prototype tree, which is a combination of a convolutional neural network (CNN) and a prototype tree with two routing path per node, as shown in Fig. 2. Here, CNN is a truncated ResNet-50, which can be represented by a mapping function f [35]. Therefore, the latent feature map of image x in m-th base learner $z_m = f(x;w_m)$ is obtained by the f, which carries the semantic feature information of each category, and is treated as the input of the prototype tree (binary tree). Here, $z_m \in R^{7\times7\times256}$, w_m denote the parameters of the CNN in the mth base learner. The binary tree consists of a leaf node set L $(l \in L)$ an internal node set N $(n \in N)$ and an edge set ξ $(e \in \xi)$. Here, l, n and e denote the leaf node, the internal node and the edge of the prototype tree, respectively. The number of leaf node l depends on the number of input classes. The number of the internal node n is determined by the height h of the prototype tree. Here, $n = 2^h - 1$. Each internal node has two optional routing branches, which is initialized as a $1 \times 1 \times 256$ latent trainable prototype. Here, the latent prototypes P_n^m of the each prototype tree are obtained in a greedy manner under

the guidance of the overall loss function [36]. In this process, the CNN parameters w_m in *m*-th branch are also obtained from the hierarchical inference process. Hence, each latent prototype $P_n^m \in P^m$ represents a routing activation pattern for the model inference process, which is close or even identical to the observations in the input image space. Where P^m denotes the latent prototype set in the *m*th subtree, P_n^m is specifically the *n*th latent prototype in *m*th base learner. It can be considered as the most distinctive feature of two class or even two clusters. The classification and inference process of the prototype tree is shown in Fig. 2.

C. Prototype Visualization

As shown in Fig. 2, the learned latent prototype P_n^m can be treated as a kernel "slides" over the all the extracted feature maps $\{z_m\}_{i=1}^N$ in *m*-th base learner and mapped into the input space according to the feature similarity. Therefore, the image prototypical parts ρ_n^{m*} is obtained to ensure the interpretability of input pathology image based on least Euclidean distance between the current image patch ρ_n^m and the latent prototype P_n^m using the minimum pooling operation:

$$\rho_n^{m*} = \underset{\rho_n^m \in \{z_m\}}{\operatorname{argmin}} ||\rho_n^m - P_n^m||_2 \tag{10}$$

Here, ρ_n^{m*} denotes the *n*-th projected image patch closest to the prototype image P_n^m in *m*-th base learner, which is also the class-specific discriminative prototypical part in the input space. Here, ρ_n^{m*} is also the visualized feature of a node in each prototype tree. In the implementation, the Euclidean distance criteria also affects the routing of arbitrarily feature representation z_m through the corresponding node *n*.

D. Inference and Prediction

In our case, the decision tree is a series of decision stumps arranged hierarchically in a top-down fashion, which means an internal node can only route either 'present' path (right) or 'absent' path (left) with a probability within the interval [0,1]. It is also known as a soft decision tree [37], [38], [39], [40]. Here, we use $e(n, n.right) \in \xi$ and $e(n, n.left) \in \xi$ to denote the routing path of present and absent, respectively. Therefore, the probability of routing sample (z_m) through the right path in *m*th base learner is defined as:

$$p_{e(n,n.right)}^m(z_m) = \exp(-||\rho_n^{m*} - P_n^m||) \tag{11}$$

Hence, the probability of routing through the left path can be written as:

$$p_{e(n,n.left)}^m = 1 - p_{e(n,n.right)}^m \tag{12}$$

As defined by the soft decision strategy, all internal nodes will be traversed through all edges (ξ) with a certain probability to reach the terminal node, and the probability of reaching the terminal node l is denoted as π_l, which is the product of all corresponding routings edges (e) :

$$\pi_l(z_m) = \prod_{e \in e_l} p_e^m(z_m) \tag{13}$$

Where e_l denotes the sequence of edges from the root node to leaf node l.

Since the soft decision tree is applied in the implementation, learning the distributions of the leaves is a global problem in each base learner. Therefore, the terminal node contains a trainable parameter $c_{m,l}$, which represents the learned class probabilities distribution of leaf l in the *m*th base learner. Thus, the softmax function σ should be applied on $c_{m,l}$ for each subtree to normalize the distribution of class probabilities in the terminal node.

For an input image *x*, the predicted class probability of *m*th base learner is obtained by traversing all edges in the tree by its latent representation $z_m = f(x|w_m)$ such that all leaves contribute to the final prediction $\hat{y}_m$. Therefore, the final prediction of the *m*th base learner can be written as:

$$\hat{y}_m(x) = \sum_{l \in L} \sigma(c_{m,l}) \cdot \pi_l(z_m) = \sum_{l \in L} \sigma(c_{m,l}) \cdot \pi_l(f(x|w_m)) \tag{14}$$

Here, the base learner updates the leaf distribution parameters $c_{m,l}$ with iterative scheme of derivative-free algorithm [37]:

$$c_{m,l}^{(t+1)} = \sum_{(x,y)} \left(\sigma\left(c_{m,l}^{(t)}\right) \odot y_m \odot \pi_{m,l}\right) \oslash \hat{y}_m \tag{15}$$

Where superscript t is the training epoch in the iteration, $c_{m,l}^{(t+1)}$ and $c_{m,l}^{(t)}$ denote the class probabilities distribution of leaf l for the *m*th base learner in epoch t and $t+1$, respectively. Here, $\{c_{m,l}^{(t)}\}_t$ are *K-dimensional* vectors. The initialized distribution $c_{m,l}^{(0)}$ can be an arbitrary value as long as every element is positive. A typical choice of $c_{m,l}^{(0)}$ is the uniform distribution in all leaves. $\odot$ is an element-wise multiplication operator. $\oslash$ represents the element-wise division operation.

Therefore, the final prediction $\hat{y}(x)$ of the EnABPT model is obtained by weighted soft voting on the normalized class probabilities of the *m* base learner $\hat{y}_m(x)$. It can be written as:

$$\hat{y}(x) = \sum_m \beta_m \cdot \text{softmax}(\hat{y}_m(x)) \tag{16}$$

E. The Overall Loss Function of the EnABPT

In our framework, both the convolutional layer connection weights $\{w_m\}$ and prototypes $\{P_n^m\}_{n=1}^n$ in each base learner are jointly optimized to obtain an ensemble predictive model including the reasoning process. Therefore, the overall loss function of each base learner is not only related to the accuracy, but also the interpretability. For image classification task, the *NML* loss in (3) is used as part of the loss function to penalize misclassified samples in the training set and improve the model accuracy. Therefore, *NML* loss can optimize both layer connection weights and prototypes of each base learner. To improve the interpretability of each sub model in the decision-making process, a regularization term is formulated as follows:

$$L_{Clus} = \sum_{n=1}^{n} \min_{\rho_n^m \in patches(f(x_i))} ||\rho_n^m - P_n^m||_2^2 \tag{17}$$

Where L_{Clus} denotes the interpretable loss, which encourages the projected prototypes to be closer to the learned prototypes P_n^m in the latent space.

The obtained model is a trade-off between accuracy and interpretability. Therefore, the overall loss of the base learner L_{total} can be written as:

$$L_{total} = L_{class} + \lambda L_{Clus} \tag{18}$$

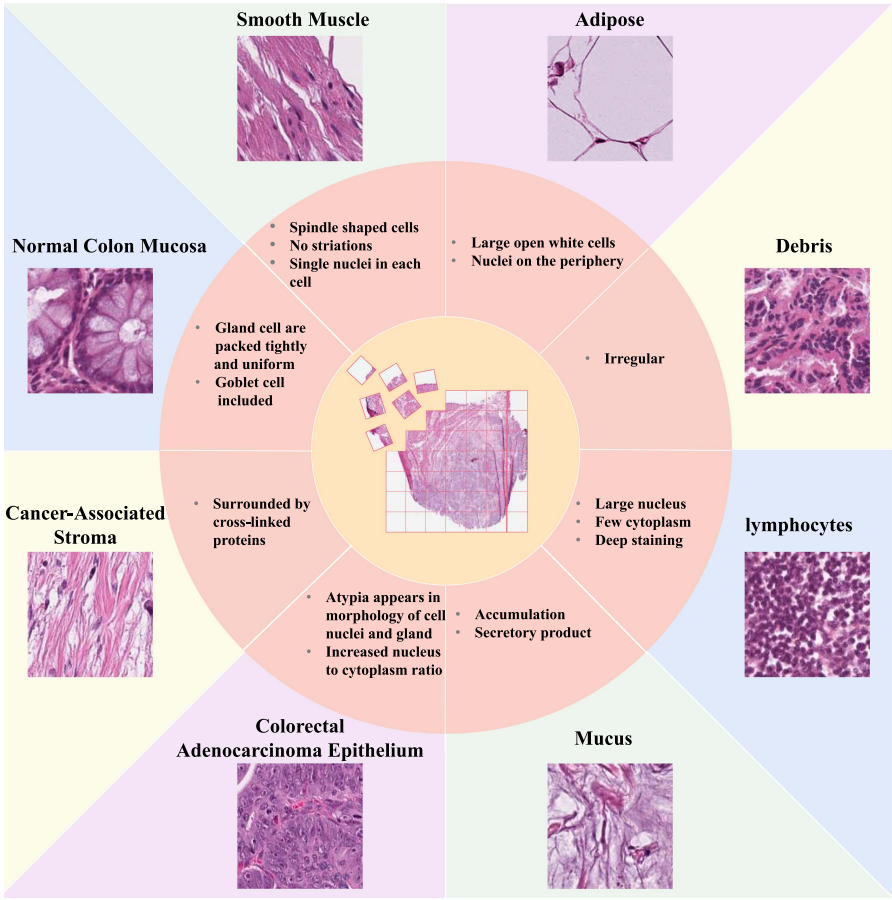

Fig. 3. Colon pathological image patches and corresponding interpretable features.

Here, λ is the coefficient of the L_{Clus}. In EnABPT, L_{total} is iteratively applied on each base learner based on the forward stage distribution algorithm to guide the ensemble model convergence.

F. Database and Implementation Details

CRCH Dataset: It is one of the largest public H&E stained pathology image datasets obtained from the National Cancer Center of Heidelberg and the Medical Center of Heidelberg University in Germany [41], [42].

The dataset contains about 100K non-overlapping human colon pathology training images and 7180 testing images. The training images were manually extracted from $N_{Train} = 86$ H&E stained human cancer tissue whole slides in the NCT Biobank. The test images were extracted from another $N_{Test} = 50$ patients with colorectal adenocarcinoma, which have non-overlap with those in the training set. This means that the test set has a shifted data distribution with that of the training dataset. These tissue images are belonging to nine distinct tissue classes including Adipose (ADI), Lymphocytes (LYM), Normal colon mucosa (NORM), Colorectal adenocarcinoma epithelium (TUM), Mucus (MUC), Smooth muscle (MUS), Debris (DEB), Cancer-associated stroma (STR) and Background (BACK). We excluded BACK as this type is meaningless for the interpretable classification process. we also made a trade-off between training time, class imbalance and model performance. Therefore, a total of 51200 non-overlapping colon tissue images are selected in the training, and 6400 images are used to test the performance of EnABPT model. Here, all the images are 224×224 pixels (112 μm ×112 μm) at ∼1 microns per pixel (MPP). They are color-normalized using Macenko's method before input into the framework. Fig. 3 shows eight main types of image patches and their interpretable features in colonic pathology whole slides.

CRC-TP Dataset: It is the first large-scale public dataset of H&E-stained human colon pathology images released by the University Hospitals of Coventry and Warwickshire (UHCW) [43], [44].

The image patches in CRC-TP dataset were manually extracted from 20 colorectal whole slide images. It contains about 280K non-overlapping image patches belonging to seven distinct tissue phenotypes including Tumor, Stroma, Complex Stroma, Muscle, Debris, Inflammatory and Benign, Here, each patch was assigned to a unique label based on the majority of its content. In this setting, 70% patches of each tissue phenotype are randomly selected for training and the remaining 30% are used for testing. Therefore, the patches may or may not belong to the same patient. We also made a trade-off between training time, class imbalance and model performance. Therefore, a total of 98000 non-overlapping colon tissue images are selected in the training, and 42000 images are used to test the performance of model. Here, each patch consists of 150×150 pixels (75μm ×75 μm, 0.5 μm/px) extracted at 20× magnification level.

In the modeling process, the number of training data in each base learner is identical, but the data distribution is adjusted by a weighted random sampler so that subsequent models can focus

TABLE I
PERFORMANCE COMPARISON OF THE PROPOSED MODEL ON CRCH DATASET

	Model	Accuracy					Accuracy	Sensitivity	AUC	F-1 Score
		Tree1	Tree2	Tree3	Tree4	Tree5				
Traditional classifier	CNN+SVM	/	/	/	/	/	0.8231	0.8231	0.8933	0.8231
CNN-based model	CNN+ GradCAM[26]	/	/	/	/	/	**0.9850**	**0.9850**	**0.9882**	**0.9850**
Decision-making algorithms	ProtoPNet[30]	/	/	/	/	/	0.9527	0.9527	0.9601	0.9527
	Prototype Tree[45]	/	/	/	/	/	0.9625	0.9625	0.9616	0.9625
Nuclei based method	MTCP[43]	/	/	/	/	/	/	/	/	0.8100-0.9700
Ensemble Models	Prototype Tree +Averaging	0.9721	0.9506	0.9627	0.9596	0.9589	0.9608	0.9608	0.9482	0.9608
	Prototype Tree +Bagging	0.9658	0.9576	0.9606	0.9570	0.9624	0.9687	0.9687	0.9577	0.9687
	EnABPT	0.9772	0.9663	0.9565	0.9533	0.9459	**0.9780**	**0.9780**	**0.9668**	**0.9780**

The values in bold indicate the optimal results, while those in bold and underlined represent the performance of our EnABPT, which is suboptimal while considering the interpretability simultaneously.

on more confusing samples. Therefore, the overall classification accuracy of the ensemble model can be improved by weighted soft voting strategy of these sequence classifiers (base learner), especially for confused samples. Meanwhile, the interpretability of the model is also improved by mutual verification of these complementary base learners. In the experiment, the number of base learner m is set to 5, which is a trade-off between accuracy, confidence and complexity of the ensemble model. The maximum height of the Adaptive Boosting Prototype Tree is initialized by defining $h = 3$ according to the image categories. The initialized learning rate is 1×10^{-5}, the batch size is 32 and the training epoch of each base learner in the experiments is set to 100. The model is implemented on RTX 3060 (12G).

IV. RESULTS AND DISCUSSION

A. Accuracy

In our EnABPT framework, not only the training samples of each base classifier but also the base classifiers themselves are adaptively weighted in an ensemble learning model for boosting. Table I shows that the classification accuracy of the ensembles adaptive boosting prototype tree can reach 0.9780 on CRCH dataset, which demonstrates the effectiveness of the proposed model. The accuracy of each base learner can be obtained as {0.9772, 0.9663, 0.9565, 0.9533, 0.9459}, and the corresponding weights of the base learners are {0.2177, 0.2072, 0.1987, 0.1929, 0.1836}, respectively. The results demonstrated that the accuracy and the weights of the base learner decrease as the training rounds increases, and the base learner with higher accuracy has the larger weight in EnABPT. This is because the base learner can iteratively increase the weight of the misclassified samples in the previous round, so that the following model could be more attentive on these samples and attempt to correct them. Therefore, although the accuracy of the sub model decreases with the number of rounds, the overall prediction accuracy can be improved evidently by ensembles of m complementary base learners with weighted soft voting strategy.

Table I also compares the performance of EnABPT with CNN + SVM, CNN-based method, SOTA decision-making algorithms and conventional ensemble learning algorithms (such as Bagging and averaging). It can be seen from Table I, the performance of CNN + SVM is not as good as other models. This is because support vector machine aims to find the hyperplane that maximizes distances between the hyperplane and the support vectors. It suffers when there is no clear margin of separation between classes. Meanwhile, SVM classifiers do not perform well on multi-class prediction problems with large datasets, especially datasets with noisy labels.

In Table I, it is intuitive that CNN-based classifiers such as CNN + gradCAM outperform the conventional ensemble learning algorithms and decision-making algorithms. This is because the design goal of a model is generally to achieve high accuracy in prediction, or high performance in a certain aspect on a given task. However, as models become more complex and sophisticated to emphasize certain aspects of performance, such as interpretability, this will come at the cost of accuracy. Here, GradCAM is a post-hoc attention mechanism, that is, it is a method for generating interpretable heat maps that can be applied to an already-trained neural network after model parameters have been fixed. Therefore, this form of interpretability does not affect the accuracy of the CNN model. However, in medical diagnosis, interpretability is an equally important factor as accuracy. This is because understanding how a model makes decisions provides an insight into why certain decisions are made and how they can be improved. However, the post-hoc interpretability obtained by gradCAM cannot provide a decision-making process, which will greatly reduce the reliability of the model. It is also untraceable when the generated heatmaps are not consistent with those of human experts. While for decision-making models and ensemble based frameworks such as EnABPT, additional interpretability constraints are introduced in the iteration to optimize both interpretability and accuracy. This will cause the degradation of the model in terms of accuracy and F-1 score. Meanwhile, in EnABPT, the decision tree of each base learner traverses the space of possible branches in a top-down greedy search manner without backtracking. Here, the greedy algorithm is based on selecting locally optimal interpretability features at each node for splitting criterion. By making these local optimal choices, the model can obtain an approximate

optimal solution globally. Therefore, the performance of EnABPT is limited because the model provides interpretability at a certain expense of accuracy. Furthermore, in boosting framework, each subsequent weak learner is forced to concentrate on the examples that are misclassified in the previous ones and attempt to correct them by intensively learning confusing samples. This means that examples that are difficult to predict receive ever-increasing influence as iterations proceed. In other words, the accuracy of EnABPT is also sensitive to the number of base learners. We can achieve higher accuracy by increasing the number of base learners. However, the computational pressure also increases with the number of base learners. Here, our EnABPT trades off model complexity, training time and interpretability while considering computational burden. Thus, CNN + gradCAM outperformed EnABPT in terms of accuracy, AUC and F-1 score when only five base learners are included in EnABPT. Whereas our model addresses the "accuracy-interpretability trade-off" issue at the cost of $\sim$1% accuracy.

While for existing SOTA decision making models such as ProtoPNet and Prototype Tree, they can provide prototypes to explain the classification results and even generate the reasoning process. However, their accuracy is only $\sim$95%, which is relatively lower than our EnABPT. This is because both models contain only a single base learner that learns the mapping function from the original data distribution. Their performance is limited without shifting the data distribution.

It is worth noting that some nucleus-based machine learning methods such as MTCP model can achieve SOTA performance in tissue phenotyping. The F-1 score of MTCP model ranged from 0.8100 to 0.9700 (depending on weighting coefficient in loss function) in classifying seven out of the eight tissue phenotypes in CRCH dataset excluding ADI tissue. However, our EnABPT still outperformed MTCP in classifying 8 types of tissue phenotypes. This is because MTCP model is based on cell detection and classification, which only considers the features of the nucleus and their interactions but ignores the meaningful information in the surrounding. Therefore, the application of the MTCP model is limited and cannot predict the tissue phenotypes without nuclei, such as ADI tissue. While our EnABPT can make predictions for all tissue phenotypes. Additionally, MTCP is a semi-supervised cellular community detection algorithm for tissue phenotyping, which integrates cell detection, classification, and cellular interaction features within a graph-based hierarchical framework. Here, the cell classification network was trained based on five specific labeled cell types including Tumor epithelial cells, Normal epithelial cells, Spindle-shaped cells, Inflammatory cells, and Necrotic cells. Therefore, the performance of MTCP is highly sensitive to its hyper parameters and prior information obtained by the cell classification network. This means that the performance of MTCP will vary on different datasets.

For ensemble learning algorithms, the results showed that the accuracy of bagging and averaging models with 5 identical neural prototype tree is 0.9687 and 0.9608, which is lower than the proposed model. This is because bagging and averaging are parallel ensemble learning frameworks, the data distribution for each base learner is nearly identical. There is no special base learner committed to focus on confusing samples, thereby the predictive ability of the model is not significantly improved by ensemble strategy. The purpose of bagging and averaging is to obtain a stable model by reducing the overall variance of the predictions. Whereas the proposed method could effectively improve the prediction accuracy and reduce the bias of whole model through sequentially focusing on poor predictions in the previous round. Therefore, EnABPT leverages the complementary properties of base learners and a weighted soft voting strategy to outperform the other two ensemble-based learning methods in terms of indicators such as accuracy, AUC and F-1 score.

Table II shows the performance of EnABPT and comparative models on the CRC-TP dataset. As shown in Table II, the classification accuracy of the EnABPT can reach 0.8389 for seven types of pathological images. The accuracy of each base learner can be obtained as {0.8241, 0.8100, 0.7968, 0.7764, 0.7637}, and the corresponding weights of the base learners are {0.2138, 0.2096, 0.2008, 0.1926, 0.1832}, respectively. In Table II, the results also demonstrate that the EnABPT model outperforms other ensemble-based models (bagging and averaging), CNN + SVM, as well as SOTA decision-making algorithms. Compared with CNN-based models, although our EnABPT model achieves slightly lower results on the CRCH dataset, it also addresses the "accuracy-interpretability trade-off" at the cost of $\sim$1.5% accuracy. Therefore, the results on CRC-TP dataset also demonstrate the effectiveness of the proposed model.

Notably, the F-1 score of the MTCP model ranged from 0.8200 to 0.9300, which outperforms our EnABPT on CRC-TP dataset under certain conditions. This is mainly because of the following two reasons: (1) As mentioned above, the MTCP is a complex semi-supervised machine learning algorithm based on a prior information of the cellular features. Whereas the prior information is obtained from a classification network based on labeling of five specific types of cell nuclei. Therefore, it is obviously that the semi-supervised approach can outperform EnABPT model in tissue phenotype prediction. (2) the MTCP is a graph-based hierarchical framework constructed on patch-level cellular interactions. This algorithm only utilizes the interaction features of the nucleus without considering the background. Thus, it is not sensitive to noisy background when prior information of the nuclei can be properly applied.

However, both semi-supervised learning approach and the realization of noisy background exclusion requires exhaustive annotations for the cells in histology images by experienced pathologists, which is not always available. In contrast, our EnABPT does not require any cellular-level annotations at all. Furthermore, it is also challenging in detection and classification of morphologically heterogeneous nuclei such as thin fibroblasts, small nuclei and overlapped nuclei. Therefore, the nucleus-based MTCP algorithm has limitations in analyzing the diversity of nuclear morphology and tissue phenotypes. While our EnABPT is not nucleus-based, which can classify tissue phenotypes of arbitrary morphology, including tissues with and without nuclei. The interpretability provided by the EnABPT is also not limited to nuclear-level features.

B. Interpretability

Fig. 4(a)–(e) visualizes the reasoning process of each base learner embedded in the EnABPT model, which faithfully explains the intrinsic classification and inference process of the whole model. The results in Fig. 4 are obtained from the built-in prototype tree (Fig. 2) following the CNN architecture in each base learner. Therefore, the feature map of the image (z_m) serves as the input of the prototype tree, which is extracted by the

TABLE II
PERFORMANCE COMPARISON OF THE PROPOSED MODEL ON CRC-TP DATASET

	Model	Accuracy					Accuracy	Sensitivity	AUC	F-1 Score
		Tree1	Tree2	Tree3	Tree4	Tree5				
Traditional classifier	CNN+SVM	/	/	/	/	/	0.7226	0.7226	0.8457	0.7226
CNN-based model	CNN+ GradCAM[26]	/	/	/	/	/	**0.8539**	**0.8539**	0.9673	**0.8539**
Decision-making algorithms	ProtoPNet[30]	/	/	/	/	/	0.8377	0.8377	0.9162	0.8377
	Prototype Tree[45]	/	/	/	/	/	0.8078	0.8078	0.9733	0.8078
Nuclei based method	MTCP[43]	/	/	/	/	/	/	/	/	0.8200-0.9300
Ensemble Models	Prototype Tree +Averaging	0.8039	0.8116	0.8096	0.8125	0.8027	0.8081	0.8081	0.9682	0.8081
	Prototype Tree +Bagging	0.8050	0.8012	0.8120	0.8017	0.8138	0.8266	0.8266	0.9744	0.8266
	EnABPT	0.8241	0.8100	0.7968	0.7764	0.7637	**0.8389**	**0.8389**	**0.9768**	**0.8389**

(The values highlighted in bold denote the optimal solutions, while those in bold and underlined represent the performance of our EnABPT, which demonstrates sub-optimal results while considering the trade-off of interpretability).

TABLE III
INTERPRETABILITY CLASSIFICATION RESULTS OF EACH PROTOTYPE TREE

	ADI	DEB	LYM	MUC	MUS	NORM	STR	TUM
1st subtree		√		√			√	√
2nd subtree		√	√			√		√
3rd subtree		√	√			√	√	
4th subtree				√	√		√	√
5th subtree	√		√	√		√		

truncated resnet-50. As is shown in Fig. 4, each subtree contains a root node, some internal nodes and more leaf nodes, which are arranged in a hierarchical structure to form a prototype tree. Both root node and internal nodes contain a prototype ρ_n^{m*} on the left and the corresponding source image on the right. This indicates that each prototype is an image patch cropped from the yellow box in the source image, which can also be regarded as the node feature or routing criterion of its children nodes. Here, the latent prototypes of the prototype tree P_n^m are obtained in a greedy manner under the guidance of the overall loss function. Then, the obtained latent prototypes are "slides" over $\{z_m\}_{i=1}^N$ and projected to the input image space to obtain class-related distinctive prototypes ρ_n^{m*} according to the feature similarity. Therefore, each prototype (marked with yellow box) can be located in different site of the corresponding source image. Meanwhile, the source image is also visualized in the prototype tree to improve the interpretability of the model from a global perspective. Each leaf node is the highest predicted probability of the specific class. The prototypes are the most distinctive parts for the two children nodes, and the input image label is predicted according to the 'presence' or 'absence' of the prototypical part. For instance, the distinctive part of NORM and TUM is the small *patch 12* according to the 1st prototypical tree, which is the benchmark for discriminating the routing path of the nodes. However, interpreting the root node and internal nodes are a bit challenging. Therefore, we mainly interpret the reasoning process of leaf nodes with local features.

Table III is a statistical analysis of the interpretable classification results of these prototype trees. As is shown in Table III, each decision tree can generate 4 human-interpretable prototypes at the root of the leaf node, which are highlighted with the yellow boxes in the 3rd row of Fig. 4(a)–(e). Here, each type of pathological image is interpreted at least once using the ensembles adaptive boosting prototype tree framework. Some of the pathological image categories are interpreted in multiple base learners. For instance, NORM and TUM are explained in three base learners. The prototypes in the three base learners are mainly localized on epithelial cells for NORM and TUM. If the atypical morphologies appear in cell nuclei and gland, as well as the increased nucleus to cytoplasm ratio, it should be interpreted as TUM, otherwise as NORM. While the prototypes of LYM in the base learner 23, and 5 includes both nucleus and cytoplasm, which means that the interpretability of LYM is characterized by a large nucleus and few cytoplasm. As can be seen from Fig. 4 and Table III, both ADI and MUS are explained in only one base learner. The prototype of ADI in 5th base learner is large open white cells with nuclei on the periphery, while the prototype of MUS in 4th base learner is the spindle shaped cells. These interpretable features are peer-reviewed by pathologists from different medical centers, and the pathologists reached a consensus on the validity of the proposed method. Finally, it can be concluded that the class-specific prototypes obtained from each subtree are perceptually relevant, and the subtrees are able to cluster the similar-looking classes.

Furthermore, the proposed model can also interpret and visualize individual predictions through a decision routing process in subtrees, which is very similar to the human reasoning process. Fig. 5 showed the decision routings path of the base learners in EnABPT based on the learned prototypical part for two randomly selected confusing TUM image patches in the test set. All of these decision-making process are based on interpretable inference by sliding learned prototypes over the test image and calculating the similarities between them in a hierarchical structure from root to leaf nodes. Here, Euclidean distance is utilized for measuring the similarity between the prototypical part and the input image patch in latent space. If the similarity probability p>0.5, the node is routing to the 'present' branch. Otherwise, routing to the 'absent' branch. For instance,

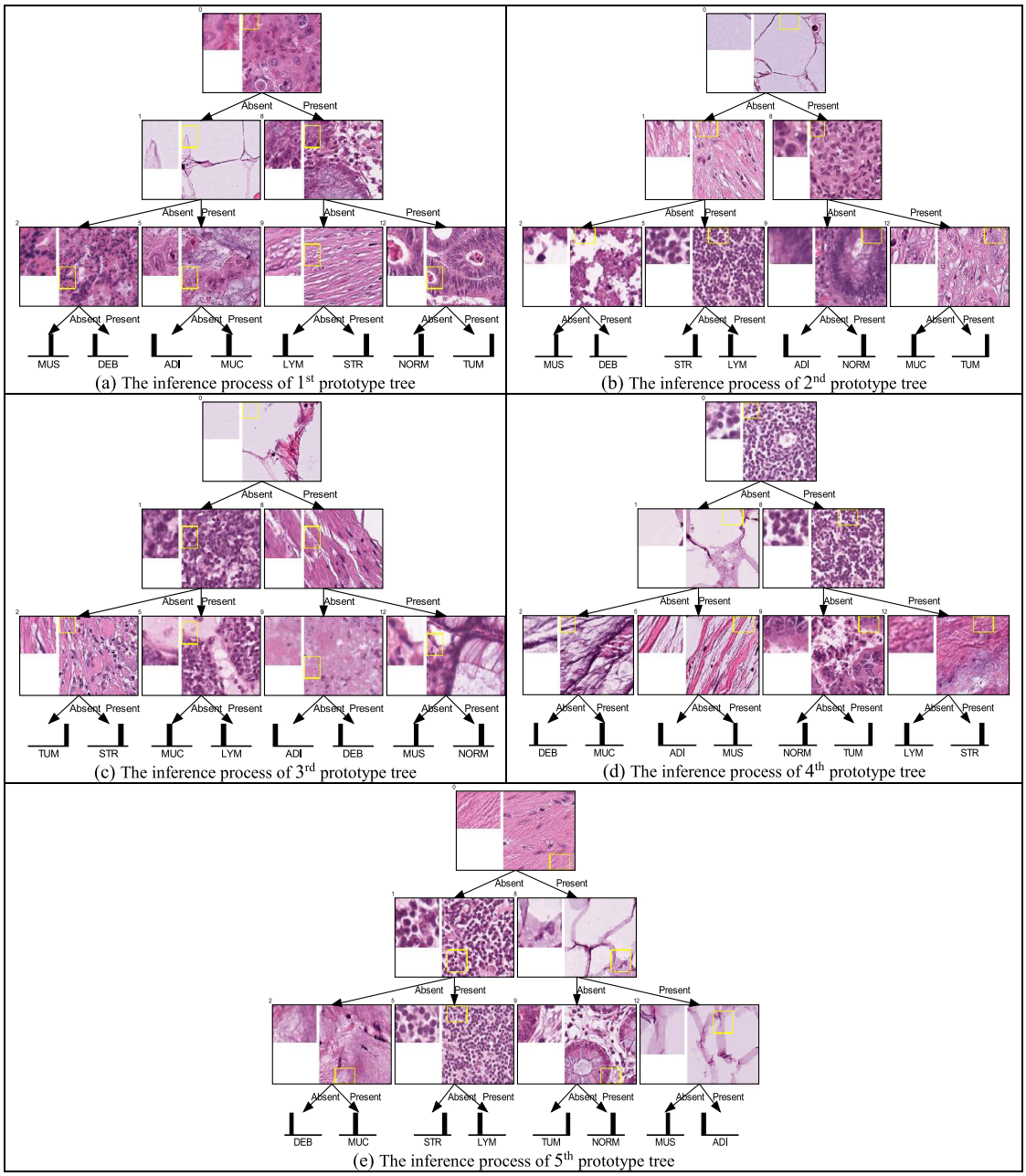

Fig. 4. Visualization of the internal decision making process of the subtrees.

NORM and TUM share some similarities in appearance. Even professional pathologists can be confused when diagnosing malignancies in early stage. The interpretive criteria are mainly depending on cellular atypia and abnormal nucleus to cytoplasm ratio. Therefore, missed diagnosed cases tend to occur in a single weak learner, as shown in Fig. 5(a), (f), (g) and (j). While other base learners attempt to correct misclassified prediction by shifting the distribution of training data, i.e., improve the identification performance by intensively learning confusing samples. Therefore, the performance of the entire model can be improved based on soft voting ensemble strategy.

As is shown in Fig. 5, it is also illustrated that the confused samples do not contain salient discriminative features, resulting in being misdiagnosed by the previous base learner in ensemble framework. Specifically, in Fig. 5(a)–(e), the TUM is partially canceration as it was in the early stage of canceration. So the 1st base learner suffers when capturing these subtle discriminative features based on the learned prototypes. Therefore, it is predicted as NORM in the 1st decision tree since the representative feature prototype of the TUM does not appear at the root of the terminal node. Meanwhile, the projected interpretable prototypes shown in Fig. 5(b) and (d) are adjacent parts in the image that contain similar content. This indicates that the feature representation for each category corresponds to a specific prototype in the latent space, which further improves the confidence of the proposed model. Fig. 5(f)–(j) are also a misclassified sample for the single base learner, which are predicted correctly by ensemble learning. From Fig. 5(f) and (g), it can be seen that the first two base learners misclassify the input sample as NORM, since the input image does not show typical features of TUM (1st base learner), and even certain features of NORM appeared, such as goblet cells (2nd base learner).

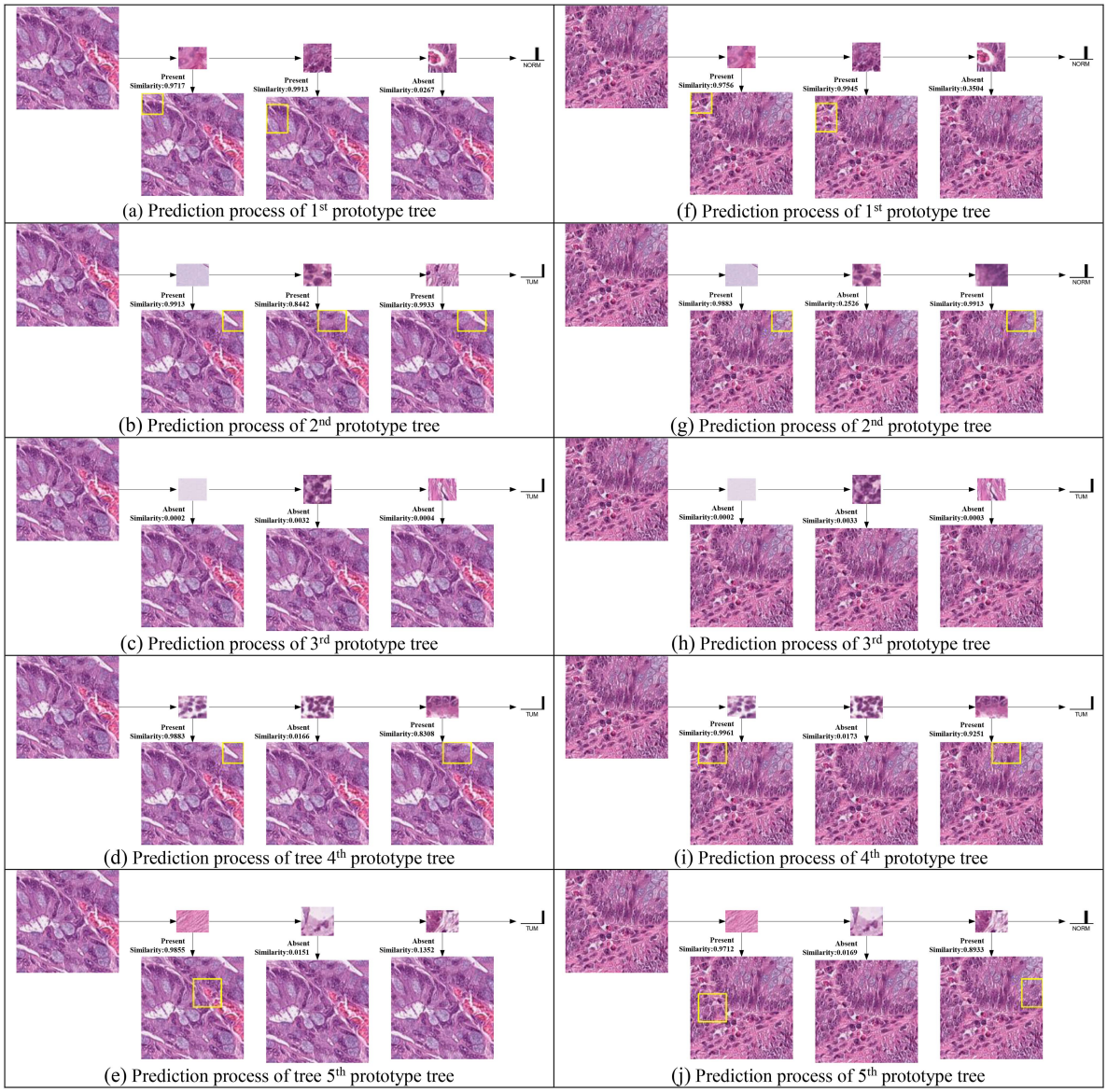

Fig. 5. Decision routing path based on local interpretability of TUM.

The prototype obtained by the inference process in Fig. 5(i) is located adjacent to that in Fig. 5(g), but the predicted outcomes are completely opposite. This is because the two prototypes have an offset in horizontal direction. The prototype in Fig. 5(i) only includes goblet cells, a statistical feature of NORM images, while the prototype in Fig. 5(g) also includes atypical epithelial cells, which is a typical feature of TUM. Therefore, it does not mean that the two reasoning processes is contradicting with each other. In Fig. 5(j), The 5th learner also misclassifies the input image as NORM for it contains the typical feature of it, as shown in the projected prototype before the output of the leaf node. From Fig. 5(f)–(j), we can conclude that the final result could be rectified and predicted correctly through EnABPT model despite only two base learners give correct predictions for this image. This is because the two base learners predict outcome with highly confidence through a well-learned class probability distribution and a soft voting ensemble strategy. Therefore, the results further demonstrate the effectiveness of the Ensembles Adaptive Boosting Prototype Tree.

V. Conclusion

To date, histopathology examination is still the golden standard of disease diagnosis, especially cancer prediction. The clinical prediction methods based on manual feature extraction suffer from long inspection cycle, labor intensity and subjective interpretation, which prone to miss the optimal chance of treatment. While the existing computer-aided diagnostic methods either focus on higher predictive performance or emphasize more interpretability. It is also a challenging task to obtain models that are both accurate and interpretable in clinical applications. Therefore, the article proposes an EnABPT framework to perform fine-grained classification for typical image patches in human colorectal pathology slides and address the "accuracy-interpretability trade-off" issue. The results showed that the EnABPT could effectively improve the prediction accuracy and reduce the bias of whole model through sequentially focusing on poor predictions in the previous base learner and striving to correct them in the following base learners. Our EnABPT

model does not require manual labeling and is very similar to human reasoning process, which opens up new perspectives for high-precise and reliable inferential diagnosis.

REFERENCES

[1] H. Sung et al., "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: Cancer J. Clinicians*, vol. 71, no. 3, pp. 209–249, 2020.

[2] R. Cao et al., "Development and interpretation of a pathomics-based model for the prediction of microsatellite instability in colorectal cancer," *Theranostics*, vol. 10, no. 24, pp. 11080–11091, 2020.

[3] E. Wulczyn et al., "Interpretable survival prediction for colorectal cancer using deep learning," *NPJ Digit. Med.*, vol. 4, no. 1, 2021, Art. no. 71.

[4] R. García-Figueiras et al., "Advanced imaging techniques in evaluation of colorectal cancer," *Radiographics*, vol. 38, no. 3, pp. 740–765, 2018.

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[6] F. Shen and G. Zeng, "Semantic image segmentation via guidance of image classification," *Neurocomputing*, vol. 330, pp. 259–266, 2019.

[7] S. Yian and S. Kyung-Shik, "Hierarchical convolutional neural networks for fashion image classification," *Expert Syst. Appl.*, vol. 116, pp. 328–339, 2019.

[8] A. Zhang et al., "Region level SAR image classification using deep features and spatial constraints," *ISPRS J. Photogrammetry Remote Sens.*, vol. 163, pp. 36–48, 2020.

[9] Y. Pei, Y. Huang, Q. Zou, X. Zhang, and S. Wang, "Effects of image degradation and degradation removal to CNN-based image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1239–1253, Apr. 2021.

[10] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3642–3649.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[12] A. J. Barnett et al., "A case-based interpretable deep learning model for classification of mass lesions in digital mammography," *Nature Mach. Intell.*, vol. 3, no. 12, pp. 1061–1070, 2021.

[13] B. C. Kwon et al., "RetainVis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records," *IEEE Trans. Visual. Comput. Graph.*, vol. 25, no. 1, pp. 299–309, Jan. 2019, doi: 10.1109/TVCG.2018.2865027.

[14] W. Hu et al., "Interpretable multimodal fusion networks reveal mechanisms of brain cognition," *IEEE Trans. Med. Imag.*, vol. 40, no. 5, pp. 1474–1483, May 2021.

[15] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *proc. 32nd AAAI Conf. Artif. Intell. 30th Innov. Appl. Artif. Intell. Conf. 8th AAAI Symp. Educ. Adv. Artif. Intell.*, 2018, pp. 1660–1669.

[16] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 782–791, doi: 10.1109/CVPR46437.2021.00084.

[17] Z. Yang, A. Zhang, and A. Sudjianto, "GAMI-Net: An explainable neural network based on generalized additive models with structured interactions," *Pattern Recognit.*, vol. 120, 2020, Art. no. 108192.

[18] Q. Zhang and S. C. Zhu, "Visual interpretability for deep learning: A survey," *Front. Inf. Technol. Electron. Eng.*, vol. 19, no. 1, pp. 27–39, 2018.

[19] D. Erhan et al., "Visualizing higher-layer features of a deep network," *Univ. Montreal*, vol. 1341, no. 3, 2009, Art. no. 1.

[20] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[22] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. Int. Conf. Learn. Representation*, Banff, Canada, 2014.

[23] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

[24] D. Smilkov et al., "Smoothgrad: Removing noise by adding noise," 2017, *arXiv:1706.03825*.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[26] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020.

[27] Q. Zhang, Y. N. Wu, and S. -C. Zhu, "Interpretable convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8827–8836.

[28] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3319–3327.

[29] O. Li et al., "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Proc. 32nd AAAI Conf. Artif. Intell. 30th Innov. Appl. Artif. Intell. Conf. 8th AAAI Symp. Educ. Adv. Artif. Intell.*, 2018, pp. 3530–3537.

[30] C. Chen et al., "This looks like that: Deep learning for interpretable image recognition," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8930–8941.

[31] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[32] M. A. Ganaie et al., "Ensemble deep learning: A review," *Eng. Appl. Artif. Intell.*, vol. 115, 2022, Art. no. 105151.

[33] A. Ferrario and R. Hämmerli, "On boosting: Theory and applications," Available at SSRN 3402687, 2019.

[34] S. S. Azmi and S. Baliga, "An overview of boosting decision tree algorithms utilizing AdaBoost and XGBoost boosting strategies," *Int. Res. J. Eng. Technol.*, vol. 7, no. 5, pp. 6867–6870, 2020.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[36] D. P. Kingma and L. J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representation*, San Diego, USA, 2015.

[37] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree," 2017, *arXiv:1711.09784*.

[38] O. Irsoy, O. T. Yildiz, and E. Alpaydin, "Soft decision trees," in *Proc. 21st Int. Conf. Pattern Recognit.*, 2012, pp. 1819–1822.

[39] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulò, "Deep neural decision forests," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1467–1475.

[40] R. Tanno et al., "Adaptive neural trees," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6166–6175.

[41] J. N. Kather et al., "Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study," *PLoS Med.*, vol. 16, no. 1, 2019, Art. no. e1002730.

[42] T. Hassan et al., "Nucleus classification in histology images using message passing network," *Med. Image Anal.*, vol. 79, 2022, Art. no. 102480.

[43] S. Javed, A. Mahmood, N. Werghi, K. Benes, and N. Rajpoot, "Multiplex cellular communities in multi-gigapixel colorectal cancer histology images for tissue phenotyping," *IEEE Trans. Image Process.*, vol. 29, pp. 9204–9219, 2020.

[44] S. Javed et al., "Cellular community detection for tissue phenotyping in colorectal cancer histology images," *Med. Image Anal.*, vol. 63, 2020, Art. no. 101696.

[45] M. Nauta, R. Van Bree, and C. Seifert, "Neural prototype trees for interpretable fine-grained image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14933–14943.