FAST: A Dual-tier Few-Shot Learning Paradigm for Whole Slide Image Classification

Kexue Fu[16*], **Xiaoyuan Luo**[23*], **Linhao Qu**[23*], **Shuo Wang**[23], **Ying Xiong**[4],
Ilias Maglogiannis[5], **Longxiang Gao**[16‡], **Manning Wang**[23‡]
[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education,
Shandong Computer Science Center (National Supercomputer Center in Jinan),
Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2]Digital Medical Research Center, School of Basic Medical Sciences, Fudan University
[3]Shanghai Key Lab of Medical Image Computing and Computer Assisted Intervention
[4] Fudan University [5] University of Piraeus
[6] Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing,
Shandong Fundamental Research Center for Computer Science, Jinan, China
{fukx, gaolx}@sdas.org, {19111010030, mnwang}@fudan.edu.cn

Abstract

The expensive fine-grained annotation and data scarcity have become the primary obstacles for the widespread adoption of deep learning-based Whole Slide Images (WSI) classification algorithms in clinical practice. Unlike few-shot learning methods in natural images that can leverage the labels of each image, existing few-shot WSI classification methods only utilize a small number of fine-grained labels or weakly supervised slide labels for training in order to avoid expensive fine-grained annotation. They lack sufficient mining of available WSIs, severely limiting WSI classification performance. To address the above issues, we propose a novel and efficient dual-tier few-shot learning paradigm for WSI classification, named FAST. FAST consists of a dual-level annotation strategy and a dual-branch classification framework. Firstly, to avoid expensive fine-grained annotation, we collect a very small number of WSIs at the slide level, and annotate an extremely small number of patches. Then, to fully mining the available WSIs, we use all the patches and available patch labels to build a cache branch, which utilizes the labeled patches to learn the labels of unlabeled patches and through knowledge retrieval for patch classification. In addition to the cache branch, we also construct a prior branch that includes learnable prompt vectors, using the text encoder of visual-language models for patch classification. Finally, we integrate the results from both branches to achieve WSI classification. Extensive experiments on binary and multi-class datasets demonstrate that our proposed method significantly surpasses existing few-shot classification methods and approaches the accuracy of fully supervised methods with only 0.22% annotation costs. All codes and models will be publicly available on `https://github.com/fukexue/FAST`.

1 Introduction

With the advent of Whole Slide Images (WSI) scanners, automated diagnosis based on WSIs has become a critical problem in the field of computational pathology [53, 48, 26]. Due to the huge size of WSI [51], deep learning-based methods typically divide it into a series of patches and apply classification models to each patch individually. Fully supervised methods annotate each patch

*Equal Contribution ‡Corresponding Author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

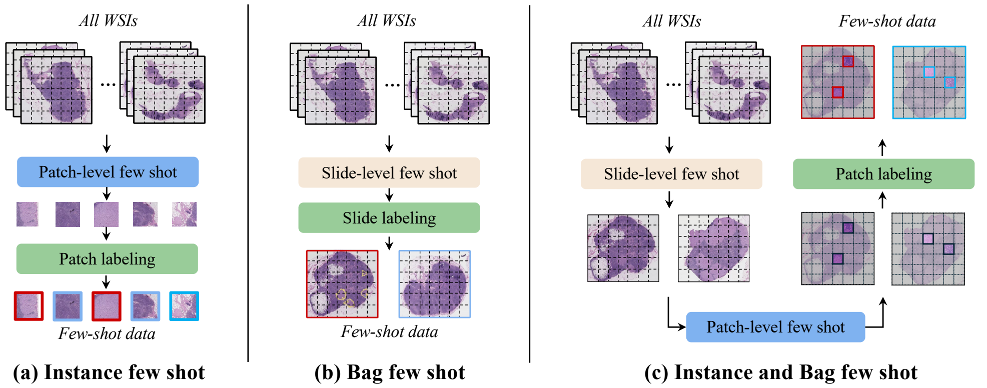

Figure 1: Different few-shot learning paradigms for WSI classification. (a) The instance few-shot method divides all WSIs into a series of patches, then selects a few samples at the patch level and annotates them at the patch level. The red box represents positive samples, and the blue box represents negative samples. (b) The bag few-shot method directly selects a few WSIs at the slide level and annotates them weakly at the slide level. (c) Our method first selects a few WSIs at the slide level, then annotates a few patches for each selected WSI. Compared to (a) and (b), our method significantly reduces annotation costs while providing patch-level supervision information.

and then train the classification model in an end-to-end manner [14, 7, 44]. However, fine-grained annotation of WSIs requires expert knowledge and is extremely expensive. To overcome these issue, weakly supervised methods formulate the WSI classification task as a multi-instance learning (MIL) problem [48, 18, 27]. In MIL, each WSI (or slide) is a bag containing thousands of instances (patches) cropped from the slide. They only use slide-level labels to train the classification model. These studies assume the availability of abundant WSIs in clinical settings. However, due to wide staining variations [30, 8], multiple cancer types [33], and rare diseases [25], many clinical scenarios can only access a limited number of WSIs. The dual obstacles of fine-grained annotation difficulties and data scarcity severely limit the available supervised information for model training. Therefore, how to avoid expensive fine-grained annotations while fully utilizing limited WSIs has become a key issue in the field of WSI classification.

Recent works [37, 61, 10, 42, 20, 54, 55, 59] in natural images have demonstrated the effectiveness of few-shot learning under limited data. Inspired by these studies, some methods [8, 6, 49, 45] gathered all patches obtained from dividing WSIs and randomly selected some patches for annotation, as shown in Figure 1(a), termed instance few shot, and then used existing few-shot learning methods from natural images for classification. These methods provide strong supervision signals for network training, but discard a large number of unlabeled patches. Our comparative experiments indicate that the methods from natural images perform poorly in few-shot WSI classification. Another methods combine weakly supervised learning with few-shot learning, such as TOP [38], which annotates only a few slide labels, as depicted in Figure 1(b), termed bag few shot. Compared to instance few shot, bag few shot methods can utilize all patches. However, slide labels belong to weak supervision signals, and the few-shot scenario leads to even greater scarcity of supervision information for MIL-based weakly supervised learning methods. Therefore, the accuracy of TOP exhibits a significant gap compared to the fully supervised learning methods [35]. Overall, the key reason for the poor performance of existing few-shot WSI classification methods is that these methods cannot simultaneously utilize strong supervision from patch labels and the remaining unlabeled patches, lacking sufficient mining of available WSIs.

In this paper, we propose a novel and efficient dual-tier **F**ew-shot learning p**A**radigm for WSI classifica**T**ion, named FAST. It consists of a annotation-efficient dual-level WSI annotation strategy and a parameter-efficient dual-branch WSI classification framework, which fully utilizes existing vision-language foundation models and can be rapidly applied to various WSI classification tasks.

Specifically, we first design a dual-level WSI annotation strategy, as shown in Figure 1(c). Under this strategy, a small number of WSIs are selected at the slide-level, followed by labeling a small number of patches within each selected WSI. Experts only need to annotate a very small number of patches without needing to perform fine-grained annotation on the entire WSI, significantly reducing the cost of fine-grained annotation while increasing the speed of annotation. Based on the proposed annotation strategy, we formulate the few-shot WSI classification task as a dual-tier few-shot learning problem. Unlike conventional few-shot learning [37], which uses an "N-way K-shot" setting, FAST's "shots" consists of two levels: bag and instance. Subsequently, under the setting of dual-tier few-shot learning, we propose a dual-branch few-shot WSI classification framework that combines vision-language (V-L) models [41, 1]. To fully utilize the prior knowledge of V-L models and the limited WSIs, the classification framework includes a cache branch and a prior branch. For the cache branch, we use the image encoder of the V-L model CLIP [41] to extract features of all patches, then construct a cache model using the labeled instances, and finally classify each instance through knowledge retrieval. However, due to the very limited number of annotated instances, the cached model has only limited knowledge and lacks generalization capability. To improve the cache model's performance, we incorporate a large number of unlabeled instances from the WSIs into the cache model, and treat their labels as learnable parameters, which effectively increases the knowledge capacity of the cache model. During training, we use annotated instances as supervision to optimize these parameters. For the prior branch, we first use GPT4-V [1] to obtain task-related prompts, and then utilize CLIP's text-image matching prior and prompt-learning techniques to design a learnable visual-language classifier. Finally, we integrate the outputs of the cache and prior branches to obtain the final classification results. This framework is based on the foundational model, requires minimal optimization of parameters (cache model and prompt vectors), and can achieve parameter-efficient fine-tuning with a small amount of WSIs and labels. Additionally, by leveraging the prior knowledge of the foundational model, FAST can maintain good accuracy and generalization even with extremely limited annotated data. These characteristics make FAST suitable for rapid adaptation to various WSI classification tasks. In summary, our main contributions are as follows:

- We propose a novel few-shot learning paradigm for WSI classification, which achieves high-accuracy WSI classification and rapid adaptation to various WSI classification tasks under low-cost annotation.
- We propose an efficient dual-level WSI annotation strategy, which can provide patch-level supervisory information at a cost close to that of slide-level annotation.
- We propose a learnable cache model based on the foundation model, which fully utilizes both annotated and unannotated patches. Furthermore, we utilize the prior knowledge of vision-language foundation models to construct a visual-language classifier, combining both to further enhance the performance of the classification framework.
- Extensive experiments demonstrate that our method achieves state-of-the-art performance on the CAMELYON16 dataset and the TCGA-RENAL dataset. In addition, compared to fully supervised methods, the annotation cost is only 0.22% of that.

2 Related Work

WSI Classification According to supervised information, WSI classification methods can be divided into two categories: fully supervised learning methods based on patch-label and weakly supervised learning methods based on slide-label. Fully supervised learning methods directly draw inspiration from supervised learning methods in natural images [23, 11, 12, 15, 58, 32]. Relevant studies on diseases such as breast cancer [44, 43], lung cancer [7, 3, 31, 14], and prostate cancer [22, 29] indicate that such methods have approached or even surpassed expert diagnostic accuracy [36]. However, expensive fine-grained annotation prevents their widespread adoption in clinical practice. Weakly supervised learning methods [48, 18, 27, 39, 40, 19, 48, 28, 35] formulate the WSI classification task as a multi-instance learning problem, avoiding expensive fine-grained annotation. Although great progress has been made, these studies rely on large amounts of training data and and cannot address the common issue of data scarcity encountered in practical clinical settings. In this paper, we propose a novel WSI classification paradigm composed of an efficient annotation strategy and a prior knowledge classification framework. The proposed method not only alleviates the poor performance issues of existing methods caused by data scarcity and difficulties in fine-grained annotation, but also enables rapid adaptation to various disease WSI classification tasks.

Few-shot Learning for WSI Classification Inspired by the success of few-shot learning in natural images, similar research has emerged in pathology images. Some studies used meta-learning methods, such as MAML [10, 42], prototypical networks [50, 9], and matching networks [54], for tasks like whole-genome doubling prediction [6] and cancer classification [49, 45]. Limited by the scale of pre-training data, these methods only achieve limited generalizability. Another group of studies borrowed the idea of fine-tuning V-L models, with related research still in the early stages. For example, CITE [62] applied visual prompt fine-tuning techniques to few-shot learning in pathology images. CLIPath [24] fine-tuned a learnable network layer on top of a frozen foundation model to transfer foundation model knowledge. However, these studies are only applicable to small-sized pathology image patches and cannot directly handle entire WSI. Qu et al. [38] leveraged the powerful generalization capabilities of CLIP, successfully achieving WSI classification tasks using only slide labels. Limited by the weakly supervised slide label, there still exists a significant gap in classification accuracy compared to fully supervised methods. PLIP [16] and CONCH [34] fine-tuned multimodal large models like CLIP [41] and CoCa [56] to perform pathology classification tasks. However, they still rely on large-scale pathology image-text pair datasets for effective performance. Different from previous works, we propose a dual-level few-shot annotation strategy and a dual-tier few-shot learning formulation approach for WSI classification, which balances annotation cost and granularity, achieving excellent classification accuracy close to fully supervised methods.

Vision-Language Model Adaptation V-L foundation models such as CLIP [41], ALIGN [21], and Florence [57] have demonstrated remarkable generalization capabilities. How to fine-tune these V-L foundation models for adaptation to downstream tasks is crucial. The popular adaptation strategies can be divided into two groups: prompt tuning and feature adapter. CoOp [63] is a representative method of prompt tuning, which optimizes a set of learnable prompt tokens to enhance the performance of V-L models in downstream tasks. A representative method of feature adapter is CLIP-Adapter [13], which fine-tunes the CLIP model by adding a lightweight residual module after the encoder. Furthermore, Tip-Adapter [61] constructs a key-value cache model to integrate the knowledge from a few-shot training set directly into the CLIP model, effectively speeding up model convergence during fine-tuning. In the field of natural images, many subsequent works based on Tip-Adapter have also made significant contributions to the development of foundation model adaptation. For example, CaFo[60] effectively combines the different prior knowledge of various pre-trained models by cascading multiple foundation models. CO3[46] goes a step further by considering both general and open-world scenarios, designing a text-guided fusion adapter to reduce the impact of noisy labels. Similarly, for open-world few-shot learning, DeIL[47] proposes filtering out less probable categories through inverse probability prediction, significantly improving performance. APE[64] proposes an adaptive prior refinement method that significantly enhances computational efficiency while ensuring high-precision classification performance. Due to the huge size and the lack of pixel-level annotations, these methods cannot effectively solve the classification problem of WSIs. However, existing methods are only applicable to fully annotated data and cannot effectively utilize unannotated patches in pathology images. Our work is inspired by Tip-Adapter, but it differs from Tip-Adapter. The key-value cache model built by Tip-Adapter only allows the key to be learnable. To fully utilize all patches in WSIs, we built a cache model where keys and values are learnable. This effectively facilitates the learning of correct label information for a large number of unlabeled patches, significantly enhancing the performance of CLIP for WSI classification.

3 Method

In this chapter, we first explain how to efficiently annotate WSIs, then describe how to formulate the few-shot WSI classification problem under the novel annotation strategy, and finally introduce the classification framework of FAST.

3.1 Annotation Strategy and Problem Formulation

For a better understanding, we first present a fully labeled WSI dataset. Given a dataset $X_{train} = \{X_1, X_2, \ldots X_M\}$ consisting of M WSIs, where each WSI $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,U_i}\}$ consisting of U patches. Each patch $x_{i,j}$ is considered an instance, and all patches in X_i form a bag. For X_i, we have a bag-level label Y_i^B, and $Y_{train}^B = \{Y_1^B, Y_2^B, \ldots Y_M^B\}$. For $x_{i,j}$, we have an instance-level label $y_{i,j}$, and $Y_{train}^I = \{\{y_{1,1}, y_{1,2}, \ldots, y_{1,U_1}\}, \{y_{2,1}, y_{2,2}, \ldots, y_{2,U_2}\}, \ldots, \{y_{M,1}, y_{M,2}, \ldots, y_{M,U_M}\}\}$. Similarly, for the testing set, the data is denoted as X_{test}, and the labels consist of Y_{test}^B and Y_{test}^I.

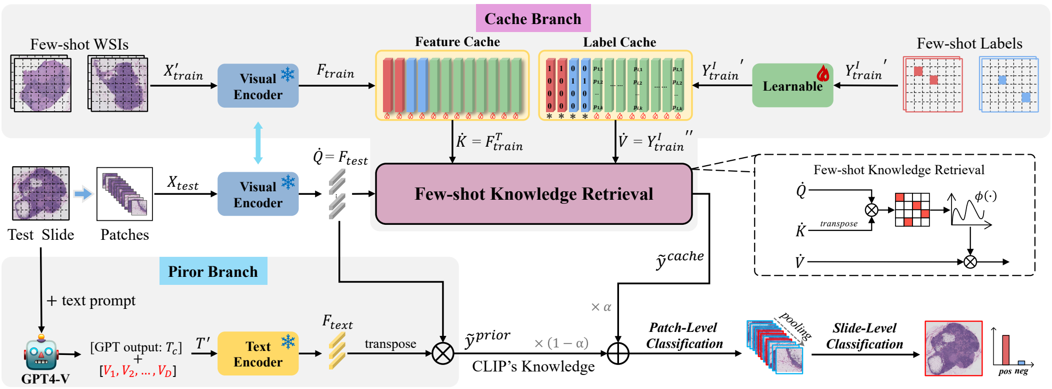

Figure 2: The structure of the FAST classification framework.

As shown in Figure 1(c), the steps of the dual-tier few-shot annotation strategy are as follows: firstly, we collect a small number of WSIs $X_{train}' = \{X_1, X_2, \dots X_K\}$, where K is much smaller than M. Then, for each WSI in X_{train}', only L patches are labeled, resulting in $Y^I_{train}{}' = \{\{y_{1,1}, y_{1,2}, \dots, y_{1,L}\}, \{y_{2,1}, y_{2,2}, \dots, y_{2,L}\}, \dots, \{y_{K,1}, y_{K,2}, \dots, y_{K,L}\}\}$, where L is much smaller than U. In this annotation strategy, the size of X_{train}' is much smaller than X_{train}, and the number of instance labels Y_{train}' is much smaller than Y_{train}.

In conventional few-shot learning, a "N-way K-shot" training set is provided, where N-way represents N categories, and K-shot represents K labeled samples. Under our dual-tier few-shot annotation strategy, the 'shot' includes bag-level few-shot and instance-level few-shot. Therefore, we formulate the few-shot WSI classification task as "N-way K-bshot & L-ishot", where K-bshot represents K bags, and L-ishot represents L labeled instances. During training, only X_{train}' and Y_{train}' are used for training, but during testing, the complete test dataset X_{test} and Y_{test} are used for testing.

3.2 Classification Framework

As shown in Figure 2, the classification framework consists of two branches: a learnable image cache branch and a CLIP prior knowledge branch. During training, only a very small number of instances need to be annotated to train the cache model and the learnable prompt tokens. During testing, the prediction results from both the image cache branch and the CLIP prior knowledge branch are integrated to obtain instance-level classification results. Finally, the instance-level classification results are pooled to yield a bag-level classification result.

1) Image Cache Branch The cache model consists of a feature cache and a label cache, and its construction method is illustrated in Figure 2. First, all patches from these WSIs are input into the image encoder for feature extraction, and the extracted features are stored in the feature cache.

$$F_{train} = \text{VisualEncoder}\left(X_{train}'\right)$$

The VisualEncoder represents the image encoder within CLIP. Simultaneously, the labels of the annotated instances are transformed into one-hot encoding and stored in the label cache. For the remaining instances without labels, we set their labels as learnable parameters P_{train} and store them in the label cache.

$$Y^I_{train} = \text{Cat}\left(Y^I_{train},{}', P_{train}\right)$$

$P_{train} = \{\{p_{1,L+1}, \dots, p_{1,L+K_1}\}, \{p_{2,L+1}, \dots, p_{2,L+K_2}\}, \dots, \{p_{i,L+1}, \dots, p_{i,L+K_i}\}\}$ represents the pseudo-labels of all unannotated instances in X_{train}', where p is a learnable high-dimensional vector. As illustrated in the few-shot knowledge retrieval module in Figure 2, through knowledge retrieval, we can obtain the prediction of the cache branch. Specifically, we employ an attention mechanism to implement the knowledge retrieval module, and treat the features and labels as key-value pairs. The Features F_{train} serve as keys, denoted by $\dot{K}$. The labels Y^I_{train} serve as values, denoted by $\dot{V}$. The feature of patch to be retrieved serve as query, denoted by $\dot{Q}$. The retrieval result is $\phi(\dot{Q}\dot{K}^T)\dot{V}$, where $\phi(\cdot) = softmax(\cdot)$.

We train the cache model using F_{train} and Y_{train}'. According to the idea of knowledge retrieval, given a query instance $x_{train} \in X_{train}$ and its encoded feature $f_{train} \in F_{train}$, and the prediction result obtained from the few-shot knowledge retrieval is $\tilde{y}^{cache} = f_{train} F_{train}^{T} Y_{train}^{I}{}''$. Subsequently, the cross-entropy between the predicted values $\tilde{y}_{i,j}^{cache}$ from the cache model and the ground truth $y_{i,j}$ is calculated as the loss function.

$$CacheLoss_{i,j} = \mathrm{CE}\left(\tilde{y}_{i,j}^{cache}, y_{i,j}\right) = \mathrm{CE}\left(f_{i,j} F_{train}^{T} Y_{train}^{I}{}'', y_{i,j}\right)$$

The learnable labels in the cache model can be optimized by minimizing the loss function. Additionally, to further optimize the feature space of CLIP, we also set F_{train} to be learnable as follow.

$$P_{train}^{*} = \min_{P_{train}} \sum_{i,j} CacheLoss_{i,j}, \quad F_{train}^{*} = \min_{F_{train}} \sum_{i,j} CacheLoss_{i,j}$$

The parameters of the VisualEncoder are frozen. It is important to note that in practical training, when there is an excess of unlabeled data, incorporating all the unlabeled instances into the cache model can exceed memory constraints. In such cases, we use K-means clustering algorithm to select representative instances to construct a core set. Only the core set is incorporated into the cache model for training and inference.

2) CLIP Prior Knowledge Branch To further leverage the prior knowledge of the vision-language foundation model, we construct a prompt-learnable vision-language instance classifier based on CLIP's text encoder. Since the feature spaces of CLIP's text encoder and image encoder are aligned, image classification can be achieved by calculating the similarity between text and images. However, unlike natural image classification, pathological image classification requires more specialized and targeted prompts. As shown in the prior branch in Figure 2, we input a small number of annotated instance images into GPT-4V, which generates descriptions of the images combining relevant pathological knowledge, forming prompts for detecting each type of WSI. Then, these category-specific prompts are feature-extracted by CLIP's text encoder,

$$f_c^{text} = \mathrm{TextEncoder}\left(T_c\right), \quad T_c = [W]_{c,1}[W]_{c,2}\ldots$$

where f_c^{text} is the text feature for category c, $f_c^{text} \in F_{text}$. T_c is the prompt encoding for category c, consisting of multiple word vectors $[W]$. Finally, given the CLIP-encoded test image feature f_{test}, the classification result under N-class text descriptions is,

$$\tilde{y}^{prior} = p\left(y = c \mid f_{test}\right) = \frac{\exp\left(\cos\left(f_c^{text}, f_{test}\right)/\tau\right)}{\sum_j^N \exp\left(\cos\left(f_j^{text}, f_{test}\right)/\tau\right)}$$

where τ is the temperature coefficient, learned by CLIP during the pre-training phase.

Furthermore, inspired by CoOp, we make the category-specific prompts learnable. Specifically, we add D learnable tokens to the prompts generated for each category, combining them into the final learnable prompts T'. The new text feature is as follows,

$$f_c^{text'} = \mathrm{TextEncoder}\left(T_c{}'\right), \quad T_i' = [W]_{i,1}[W]_{i,2}\ldots[V]_{i,1}[V]_{i,2}\ldots[V]_{i,D}$$

Based on the new text feature and the image feature $f_{i,j}$, the classification result of prior branch is,

$$\tilde{y}_{i,j}^{prior} = p\left(y_{i,j} = c \mid f_{i,j}\right) = \frac{\exp\left(\cos\left(f_c^{text'}, f_{i,j}\right)/\tau\right)}{\sum_j^N \exp\left(\cos\left(f_j^{text'}, f_{i,j}\right)/\tau\right)}$$

Similar to the image cache branch, we also use all annotated instances to train the prior knowledge branch. The optimization function and loss function of the prior branch are as follows,

$$[V] = \min_{[V]} \sum_{i,j} TextLoss_{i,j}, \quad \text{where } TextLoss_{i,j} = \mathrm{CE}\left(\tilde{y}_{i,j}^{prior}, y_{i,j}\right)$$

The parameters of the TextEncoder are also frozen. Finally, we combine the predictions and loss functions from the image cache branch and the prior knowledge branch to obtain the overall model's instance-level classification result and loss function,

$$\tilde{y} = \alpha \cdot \tilde{y}^{cache} + (1-\alpha) \cdot \tilde{y}^{prior}$$

$$Loss = \sum_{i,j} CacheLoss_{i,j} + \sum_{i,j} TextLoss_{i,j}$$

where α is the fusion weight of the two branches, which can be considered a hyperparameter. We divide the fusion weight α into equal intervals with a step size of 100, then sequentially calculate the classification accuracy for each fusion ratio, and finally select the fusion weight that yields the highest classification accuracy as the fusion weight α for this task. Since the entire network has few parameters and fast inference speed, this parameter can be quickly optimized through grid search to obtain the optimal value.

4 Experiments

4.1 Datasets and Few-Shot Scenario Simulation

We evaluated our method on the public WSI datasets CAMELYON16 [4] and TCGA-RENAL [17]. CAMELYON16 is a binary dataset used to detect whether it contains breast cancer metastasis. TCGA-RENAL is a multi-class dataset for classifying renal cancer subtypes, covering clear cell renal cell carcinoma (ccRCC), papillary renal cell carcinoma (pRCC), and chromophobe renal cell carcinoma (chRCC). For more details about the datasets, please refer to the supplementary material A.1. For few-shot scenario simulation, we first simulate the collection of few-shot WSI data in clinical settings by sampling 1, 2, 4, 8, and 16 WSIs for each category. Then, we simulate the process of expert annotation of few-shot instances within sampled bags, initially selecting 10% instances as a core set by K-means clustering [2], then randomly sampling 16 instances per category from the core set as labeled instances, with others remaining unlabeled. Through this simulation, for CAMELYON16 and TCGA-RENAL datasets, we obtained few-shot training datasets with 1, 2, 4, 8, and 16 bag shots and 16 instance shots. Considering the randomness in few-shot learning, we conducted five random samplings and model trainings for all scenarios and reported the mean and variance of classification results.

4.2 Comparing Methods and Evaluation Metrics

First, we compared FAST with zero-shot learning method CLIP [36] and fully supervised method using all instance-level labels [35], which can respectively be seen as the lower and upper bounds of deep learning method performance in few-shot WSI classification. Subsequently, due to lack of effective few-shot WSI learning methods, we compared the latest few-shot learning methods Tip-Adapter and Tip-Adapter-F [61] in natural images. Bag-level weakly supervised multi-instance learning methods, such as R2T[52], generally perform worse than instance-level fully supervised methods due to the lack of fine-grained labels. Therefore, this paper does not directly compare with multi-instance learning methods. For implementation details of our method, please refer to the supplementary material A.2. To provide a comprehensive evaluation of these methods, we reported instance-level Area Under Curve and bag-level Area Under Curve (AUC) [5]. Since the TCGA-RENAL dataset is a multi-class classification task, we reported instance-level and bag-level AUCs separately for each category.

4.3 Experimental Results

CAMELYON16 The few-shot WSI classification results on the CAMELYON16 dataset are shown in Table 1. **Firstly**, in the zero-shot setting, the instance-level AUC and bag-level AUC of zero-shot CLIP are very low, almost unable to handle the classification task. In contrast, under extreme few-shot setting with 1 bag shot and 16 instance shots, FAST achieves 0.84 instance-level AUC, which shows a significant improvement over zero-shot CLIP. **Secondly**, as the training samples increase, the performance of FAST shows a significant improvement trend. Meanwhile, our proposed method FAST consistently outperforms the comparison methods Tip-Adapter and Tip-Adapter-F across different numbers of samples. Although Tip-Adapter and Tip-Adapter-F have achieved significant success in natural images, their performance is poor on few-shot WSI classification task, especially with bag-level AUC generally below 0.6. We think there are two main reasons for this: (1) there are domain differences between pathology images and natural images, making it difficult for the text branch of CLIP to accurately classify instances and even more difficult to classify bags. (2) The limited WSIs make it more difficult to ensure the diversity of instances sampled from these WSIs, resulting in poor generalization of these methods. In contrast, while FAST also utilizes a small number of bags, it fully leverages the labeled and unlabeled instances within these bags through a learnable label cache, enabling it to learn more comprehensive instance representations. **Finally**,

Table 1: Results on CAMELYON16 dataset

Bag Shot	Instance Shot	Methods	Instance-level AUC	Bag-level AUC
0	0	Zero-shot CLIP	0.6711	0.5409
0	0	Zero-shot PLIP	0.6004	0.5434
0	0	Zero-shot CONCH	0.8929	0.7113
1	16	**FAST**	**0.8400±0.0335**	**0.6933±0.0846**
		Tip-Adapter-F	0.7162±0.0435	0.5653±0.0604
		Tip-Adapter	0.6275±0.0777	0.498±0.0187
2	16	**FAST**	**0.8584±0.0380**	**0.7595±0.0391**
		Tip-Adapter-F	0.7200±0.0595	0.5748±0.0537
		Tip-Adapter	0.6198±0.0823	0.5141±0.0156
4	16	**FAST**	**0.8864±0.0563**	**0.7359±0.0853**
		Tip-Adapter-F	0.6990±0.0890	0.5731±0.0401
		Tip-Adapter	0.5601±0.0772	0.5321±0.0152
8	16	**FAST**	**0.9060±0.0074**	**0.7742±0.0249**
		Tip-Adapter-F	0.7392±0.0180	0.6045±0.0044
		Tip-Adapter	0.6782±0.0166	0.4880±0.0097
16	16	**FAST**	**0.9151±0.0200**	**0.8197±0.0474**
		Tip-Adapter-F	0.7227±0.0098	0.5965±0.0243
		Tip-Adapter	0.6835±0.0135	0.4913±0.0164
All	All	Fully Supervised	0.9532	0.8555

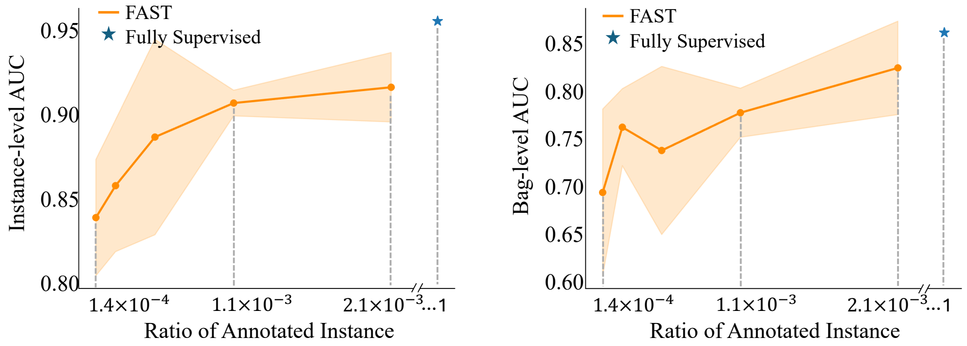

Figure 3: Results of FAST on CAMELYON16 dataset under different annotation ratio.

in the setting where only 16 bag shots and 16 instance shots are available, our proposed FAST method achieves 0.9151 instance-level and 0.8197 bag-level AUC, which is close to the performance of the fully supervised method using all instance annotations. Importantly, the annotation cost of FAST is only 0.22% of that of the fully supervised method, as detailed in Section 4.4. **In summary**, FAST achieves accuracy close to the fully supervised method with extremely low data collection and annotation costs. This significantly enhances the efficiency of establishing deep learning models in clinical settings and opens up possibilities for widespread clinical adoption of WSI classification algorithms based on deep learning.

TCGA-RENAL In the few-shot experiments on the TCGA-RENAL dataset, the classification results are shown in Table 2. Compared to the binary classification task on the CAMELYON16 dataset, the three-class classification task is more challenging. Therefore, the zero-shot CLIP method only achieves around 0.5 instance-level AUC, almost unable to handle the classification task. In the setting with very few shots, such as one or two bags, the results of all methods are relatively poor, with instance-level AUC and bag-level AUC both less than 0.7000. However, our proposed FAST still achieves SOTA performance on most metrics. As the bag shot reaches 4 or more, the classification results begin to significantly improve, and FAST also outperforms other methods on all metrics. Especially in 16 bag shots, FAST significantly outperforms the comparison methods. The average bag-level AUC for chRCC reaches 0.9234, differing by only 0.0033 from the fully supervised method. Similarly, the bag-level AUC for ccRCC and pRCC also reach 0.9254 and 0.9216 respectively, differing by only 0.0218 and 0.036 from the fully supervised method. This experimental result further demonstrates that, in complex multi-class classification tasks, FAST can approach the fully supervised method with very low annotation costs, achieving the state-of-the-art performance in few-shot learning.

Table 2: Results on TCGA-RENAL dataset

Bag Shot	Instance Shot	Methods	Instance-level AUC			Bag-level AUC			
			ccRCC	pRCC	chRCC	ccRCC	pRCC	chRCC	mean
0	0	Zero-shot CLIP	0.5475	0.5521	0.3640	0.4959	0.5192	0.4811	0.4987
0	0	Zero-shot PLIP	0.5396	0.6006	0.4774	0.6036	0.6610	0.4905	0.5850
0	0	Zero-shot CONCH	0.8127	0.9122	0.9131	0.9039	0.8936	0.9449	0.9141
1	16	**FAST**	**0.5935±0.0488**	**0.6853±0.0443**	**0.6548±0.0760**	**0.6067±0.0833**	**0.6921±0.0416**	0.6488±0.0984	**0.6492**
		Tip-Adapter-F	0.5710±0.0387	0.6533±0.0566	0.6230±0.0967	0.5981±0.0393	0.6523±0.0684	**0.6948±0.1113**	0.6484
		Tip-Adapter	0.5874±0.0503	0.6308±0.0655	0.6017±0.0734	0.5951±0.0602	0.6515±0.0668	0.6624±0.1099	0.6363
2	16	**FAST**	**0.6245±0.0733**	**0.6998±0.0229**	**0.6987±0.0582**	**0.6745±0.0900**	0.7327±0.0279	0.7329±0.0483	**0.7133**
		Tip-Adapter-F	0.6084±0.0609	0.6820±0.0405	0.6985±0.0481	0.6359±0.1143	**0.7391±0.0331**	**0.7481±0.0794**	0.7077
		Tip-Adapter	0.6068±0.0524	0.6413±0.0554	0.6238±0.0522	0.6545±0.0931	0.6682±0.1034	0.7405±0.0763	0.6877
4	16	**FAST**	**0.7107±0.1056**	**0.7547±0.0544**	**0.7652±0.0645**	**0.7684±0.1681**	**0.8260±0.0816**	**0.8143±0.1080**	**0.8029**
		Tip-Adapter-F	0.6587±0.0858	0.7266±0.0756	0.7488±0.0555	0.7220±0.1091	0.7621±0.0985	0.7132±0.1631	0.7324
		Tip-Adapter	0.6361±0.0737	0.6797±0.0820	0.6835±0.0805	0.6671±0.1384	0.7274±0.0391	0.7856±0.0866	0.7267
8	16	**FAST**	**0.7940±0.0522**	**0.8228±0.0275**	**0.8398±0.0194**	**0.8955±0.0453**	**0.8978±0.0478**	**0.8964±0.0485**	**0.8966**
		Tip-Adapter-F	0.7249±0.0529	0.7832±0.0255	0.7918±0.0239	0.7854±0.1382	0.8239±0.0595	0.7879±0.0595	0.7991
		Tip-Adapter	0.6839±0.0567	0.7552±0.0275	0.7623±0.0380	0.7735±0.0793	0.7787±0.0526	0.7908±0.0735	0.7810
16	16	**FAST**	**0.8252±0.0428**	**0.8420±0.0126**	**0.8609±0.0165**	**0.9254±0.0206**	**0.9216±0.0233**	**0.9234±0.0184**	**0.9235**
		Tip-Adapter-F	0.7409±0.0736	0.8026±0.0151	0.8030±0.0268	0.8612±0.0726	0.8235±0.0547	0.8531±0.0591	0.8459
		Tip-Adapter	0.6911±0.0484	0.7886±0.0230	0.7851±0.0191	0.7624±0.0931	0.7593±0.0400	0.8877±0.0621	0.8031
All	All	Fully Supervised	0.8917	0.9055	0.8977	0.9472	0.9576	0.9267	0.9438

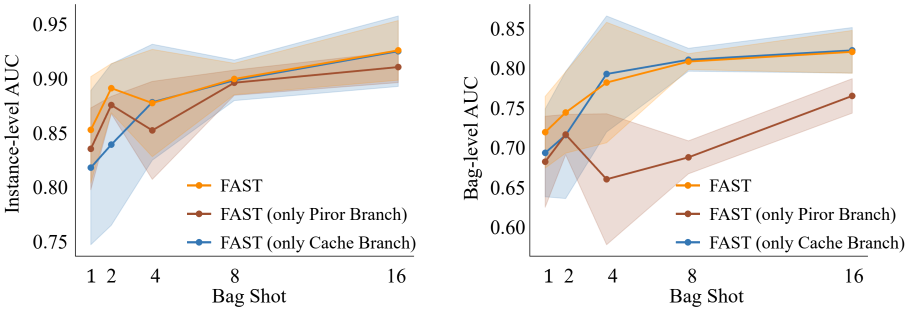

Figure 4: Comparison of cache branch and prior branch in FAST.

4.4 Annotation Efficiency

To illustrate the annotation efficiency of FAST, we compared the classification results of our method and the fully supervised method under different annotation ratios. For the CAMELYON16 dataset, the results are presented in Figure 3. It can be observed that the classification AUC of FAST rapidly increase with the growth of bag shots. When the bag shot reaches 16, the annotation only accounts for 0.22%. The average bag-level AUC reaches 96.32% of the fully supervised method. This result fully demonstrates the advantage of FAST in annotation efficiency. For the results on the TCAG-RENAL dataset, please refer to the supplementary material A.3.

4.5 Ablation Studies

To analyze the importance of each component in FAST, we conducted a serise of experiments under the conditions of 16 bag shots and 16 instance shots on the CAMELYON16 dataset, including whether to use a cache branch, whether to set the feature cache as learnable, whether to use a prior branch, and whether to utilize unlabeled instances to construct a learnable label cache. The results of the ablation experiments are shown in Table 3. The first three rows of the table correspond to zero-shot CLIP, Tip-Adapter, and Tip-Adapter-F, respectively. The next three rows correspond to FAST using only the prior branch, FAST using only the cache branch, and the complete FAST. **Firstly**, the first three rows of the table show that the instance-level AUC gradually increases from 0.6711 to 0.7277 for the three methods. This indicates that the cache branch can leverage a small amount of supervised information, and making the cache feature learnable can further optimize the feature space distribution of the cache model. However, even the best model's instance-level AUC is only 0.7227, and the pooled bag-level AUCs are all less than 0.6. **Next**, from the last three rows of the table, even with only the prior branch, using our proposed prompt-based method, the instance-level and bag-level classification AUCs reach 0.8739 and 0.7931, respectively, which outperformed the results of Tip-Adapter-F. The instance-level and bag-level AUCs of only using the cache branch can reach 0.9165 and 0.8183, respectively, demonstrating that the cache branch is crucial for FAST. By further comparing the third and fifth rows of the table, we can infer the primary reason why FAST's cache model surpasses that of

Table 3: Ablation study of FAST on CAMELYON16 dataset

Cache Branch	Learnable Feature Cache	Prior Branch	Learnable Label Cache	Instance-level AUC	Bag-level AUC
				0.6711	0.5409
✓				0.6835±0.0135	0.4913±0.0164
✓	✓			0.7227±0.0098	0.5965±0.0243
		✓		0.8739±0.0161	0.7931±0.0247
✓	✓		✓	**0.9165±0.0213**	0.8183±0.0560
✓	✓	✓	✓	0.9151±0.0200	**0.8197±0.0474**

Tip-Adapter-F in natural images. Specifically, FAST utilizes a large number of unannotated instances in a small number of bags through the learnable label cache, thereby significantly improving the capacity and generalization ability of the cache model. Finally, comparing the last two rows of the table, we find that when both cache and prior branches are utilized, FAST does not show significant improvement over using only the cache branch. This is because the number of samples is sufficient, and the prior branch primarily plays a role when there are fewer samples.

To verify the role of two branchs, we compared the prior branch and cache branch of FAST under different bag shots. The results are shown in Figure 4. When there are only 1 or 2 bags, the instance classification results of the prior branch are significantly higher than those of the cache branch. The instance and bag classification results combined with both the cache and prior branches also surpass those of using each branch separately, indicating that the prior branch performs better in extreme samples, and the information learned by the prior branch and the cache branch is complementary. As the number of bags increases to 4 or more, the results of the cache branch gradually surpass those of the prior branch. Therefore, in extreme few-shot scenarios, FAST is dominated by the prior branch, but as the sample size gradually increases, FAST is dominated by the image branch. This experimental analysis can provide effective guidance for the practical application of FAST. Additionally, we provide further analysis experiments on the number of annotated instance and the number of core sets in the supplementary material A.3.

5 Conclusion

In this paper, we propose a dual-tier few-shot learning paradigm FAST for WSI classification, which achieves low-cost WSI annotation, high-accuracy WSI classification, and adaptation to various WSI classification tasks. To achieve these goals, we introduce two key technologies in FAST, including a dual-level few-shot annotation strategy and a dual-branch classification framework. The dual-level few-shot annotation strategy effectively alleviates the problem of fine-grained annotation difficulty in WSI by annotating only a small number of patches in a limited number of WSIs. Experimental results demonstrate that the annotation cost of our method is only 0.22% of patch-level full annotation. In the dual-branch classification framework, we construct a cache branch where both features and labels are learnable, fully exploiting the partially annotated data obtained from the dual-level few-shot annotation strategy. Furthermore, combining a vision-language foundation model and prompt tuning technology, we build a prior knowledge branch to assist the cache branch in improving classification performance. Through extensive experiments, we demonstrate that FAST can achieve state-of-the-art performance on both binary and multi-class classification tasks. However, our method still has certain limitations. For bag-level classification, we simply use pooled instance classification results, ignoring the relationships between instances. In the future, we will further explore how to use instance-level classification results to obtain a better bag-level classification result.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key R&D Program of China under Grant No. 2022ZD0116800, and the National Natural Science Foundation of China under Grant 62471149 and Taishan Scholars Program under Grant TSQN202211214.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Khaled Alsabti, Sanjay Ranka, and Vineet Singh. An efficient k-means clustering algorithm. 1997.

[3] Neha Baranwal, Preethi Doravari, and Renu Kachhoria. Classification of histopathology images of lung cancer using convolutional neural network (cnn). In *Disruptive Developments in Biomedical Applications*, pages 75–89. CRC Press, 2022.

[4] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

[5] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[6] Sherry Chao and David Belanger. Generalizing few-shot classification of whole-genome doubling across cancer types. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3382–3392, 2021.

[7] Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L Moreira, Narges Razavian, and Aristotelis Tsirigos. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature medicine*, 24(10):1559–1567, 2018.

[8] Jessica Deuschel, Daniel Firmbach, Carol I Geppert, Markus Eckstein, Arndt Hartmann, Volker Bruns, Petr Kuritcyn, Jakob Dexl, David Hartmann, Dominik Perrin, et al. Multi-prototype few-shot learning in histopathology. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 620–628, 2021.

[9] Jessica Deuschel, Daniel Firmbach, Carol I Geppert, Markus Eckstein, Arndt Hartmann, Volker Bruns, Petr Kuritcyn, Jakob Dexl, David Hartmann, Dominik Perrin, et al. Multi-prototype few-shot learning in histopathology. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 620–628, 2021.

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[11] Junyu Gao, Mengyuan Chen, and Changsheng Xu. Vectorized evidential learning for weakly-supervised temporal action localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15949 – 15963, 2023.

[12] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Learning to model relationships for zero-shot video classification. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3476–3491, 2020.

[13] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

[14] Le Hou, Dimitris Samaras, Tahsin M Kurc, Yi Gao, James E Davis, and Joel H Saltz. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2424–2433, 2016.

[15] Yufan Hu, Junyu Gao, Jianfeng Dong, Bin Fan, and Hongmin Liu. Exploring rich semantics for open-set action recognition. *IEEE Transactions on Multimedia*, 2023.

[16] Zhi Huang, Federico Bianchi, Mert Yuksekgonul, Thomas J Montine, and James Zou. A visual–language foundation model for pathology image analysis using medical twitter. *Nature medicine*, 29(9):2307–2316, 2023.

[17] Carolyn Hutter and Jean Claude Zenklusen. The cancer genome atlas: creating lasting value beyond its data. *Cell*, 173(2):283–285, 2018.

[18] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

[19] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

[20] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11719–11727, 2019.

[21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[22] Davood Karimi, Guy Nir, Ladan Fazli, Peter C Black, Larry Goldenberg, and Septimiu E Salcudean. Deep learning-based gleason grading of prostate cancer from histopathology images—role of multiscale decision aggregation and data augmentation. *IEEE journal of biomedical and health informatics*, 24(5):1413–1426, 2019.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[24] Zhengfeng Lai, Zhuoheng Li, Luca Cerny Oliveira, Joohi Chauhan, Brittany N Dugger, and Chen-Nee Chuah. Clipath: Fine-tune clip with visual feature fusion for pathology image analysis towards minimizing data collection efforts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop*, pages 2374–2380, 2023.

[25] Junghwan Lee, Cong Liu, Junyoung Kim, Zhehuan Chen, Yingcheng Sun, James R Rogers, Wendy K Chung, and Chunhua Weng. Deep learning for rare disease: A scoping review. *Journal of Biomedical Informatics*, 135:104227, 2022.

[26] Bin Li, Yin Li, and Kevin W Eliceiri. 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*, pages 14318–14328, 2021.

[27] Hang Li, Fan Yang, Yu Zhao, Xiaohan Xing, Jun Zhang, Mingxuan Gao, Junzhou Huang, Liansheng Wang, and Jianhua Yao. Dt-mil: deformable transformer for multi-instance learning on histopathological image. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pages 206–216. Springer, 2021.

[28] Hang Li, Fan Yang, Yu Zhao, Xiaohan Xing, Jun Zhang, Mingxuan Gao, Junzhou Huang, Liansheng Wang, and Jianhua Yao. Dt-mil: deformable transformer for multi-instance learning on histopathological image. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pages 206–216. Springer, 2021.

[29] Wenyuan Li, Jiayun Li, Karthik V Sarma, King Chung Ho, Shiwen Shen, Beatrice S Knudsen, Arkadiusz Gertych, and Corey W Arnold. Path r-cnn for prostate cancer diagnosis and gleason grading of histological images. *IEEE transactions on medical imaging*, 38(4):945–954, 2018.

[30] Sylwia Libard, Dijana Cerjan, and Irina Alafuzoff. Characteristics of the tissue section that influence the staining outcome in immunohistochemistry. *Histochemistry and Cell Biology*, 151:91–96, 2019.

[31] Min Liu, Lanlan Hu, Ying Tang, Chu Wang, Yu He, Chunyan Zeng, Kun Lin, Zhizi He, and Wujie Huo. A deep learning method for breast cancer classification in the pathology images. *IEEE Journal of Biomedical and Health Informatics*, 26(10):5025–5032, 2022.

[32] Shaolei Liu, Xiaoyuan Luo, Kexue Fu, Manning Wang, and Zhijian Song. A learnable self-supervised task for unsupervised domain adaptation on point cloud classification and segmentation. *Frontiers of Computer Science*, 17(6):176708, 2023.

[33] Lawrence A Loeb, Keith R Loeb, and Jon P Anderson. Multiple mutations and cancer. *Proceedings of the National Academy of Sciences*, 100(3):776–781, 2003.

[34] Ming Y Lu, Bowen Chen, Drew FK Williamson, Richard J Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, et al. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024.

[35] Xiaoyuan Luo, Linhao Qu, Qinhao Guo, Zhijian Song, and Manning Wang. Negative instance guided self-distillation framework for whole slide image analysis. *IEEE Journal of Biomedical and Health Informatics*, 2023.

[36] Muhammad Khalid Khan Niazi, Anil V Parwani, and Metin N Gurcan. Digital pathology and artificial intelligence. *The lancet oncology*, 20(5):e253–e261, 2019.

[37] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7229–7238, 2018.

[38] Linhao Qu, Kexue Fu, Manning Wang, Zhijian Song, et al. The rise of ai language pathologists: Exploring two-level prompt learning for few-shot weakly-supervised whole slide image classification. *Advances in Neural Information Processing Systems*, 36, 2024.

[39] Linhao Qu, Xiaoyuan Luo, Shaolei Liu, Manning Wang, and Zhijian Song. Dgmil: Distribution guided multiple instance learning for whole slide image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 24–34. Springer, 2022.

[40] Linhao Qu, Manning Wang, Zhijian Song, et al. Bi-directional weakly supervised knowledge distillation for whole slide image classification. *Advances in Neural Information Processing Systems*, 35:15368–15381, 2022.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[42] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.

[43] Alexander Rakhlin, Alexey Shvets, Vladimir Iglovikov, and Alexandr A Kalinin. Deep convolutional neural networks for breast cancer histology image analysis. In *Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018, Proceedings 15*, pages 737–744. Springer, 2018.

[44] Kaushiki Roy, Debapriya Banik, Debotosh Bhattacharjee, and Mita Nasipuri. Patch-based system for classification of breast histology images using deep learning. *Computerized Medical Imaging and Graphics*, 71:90–103, 2019.

[45] Nazim N Shaikh, Kamil Wasag, and Yao Nie. Artifact identification in digital histopathology images using few-shot learning. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–4. IEEE, 2022.

[46] Shuai Shao, Yu Bai, Yan Wang, Baodi Liu, and Bin Liu. Collaborative consortium of foundation models for open-world few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4740–4747, 2024.

[47] Shuai Shao, Yu Bai, Yan Wang, Baodi Liu, and Yicong Zhou. Deil: Direct-and-inverse clip for open-world few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28505–28514, 2024.

[48] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems*, 34:2136–2147, 2021.

[49] Rishav Singh, Vandana Bharti, Vishal Purohit, Abhinav Kumar, Amit Kumar Singh, and Sanjay Kumar Singh. Metamed: Few-shot medical image classification using gradient-based meta-learning. *Pattern Recognition*, 120:108111, 2021.

[50] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[51] Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep neural network models for computational histopathology: A survey. *Medical image analysis*, 67:101813, 2021.

[52] Wenhao Tang, Fengtao Zhou, Sheng Huang, Xiang Zhu, Yi Zhang, and Bo Liu. Feature re-embedding: Towards foundation model-level performance in computational pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11343–11352, 2024.

[53] Jeroen Van der Laak, Geert Litjens, and Francesco Ciompi. Deep learning in histopathology: the path to the clinic. *Nature medicine*, 27(5):775–784, 2021.

[54] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[55] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8808–8817, 2020.

[56] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*.

[57] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[58] Mingzhi Yuan, Xiaoshui Huang, Kexue Fu, Zhihao Li, and Manning Wang. Boosting 3d point cloud registration by transferring multi-modality knowledge. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11734–11741. IEEE, 2023.

[59] Mingzhi Yuan, Ao Shen, Kexue Fu, Jiaming Guan, Yingfan Ma, Qin Qiao, and Manning Wang. Proteinmae: masked autoencoder for protein surface self-supervised learning. *Bioinformatics*, 39(12):btad724, 2023.

[60] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15211–15222, 2023.

[61] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pages 493–510. Springer, 2022.

[62] Yunkun Zhang, Jin Gao, Mu Zhou, Xiaosong Wang, Yu Qiao, Shaoting Zhang, and Dequan Wang. Text-guided foundation model adaptation for pathological image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 272–282. Springer, 2023.

[63] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[64] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2605–2615, 2023.

A Supplemental Material

A.1 Datasets

CAMELYON16 CAMELYON16 [4] contains 400 WSIs of lymph nodes, used to detect the presence of metastatic breast cancer. WSIs containing metastases are labeled positive, while others are negative, with pixel-level annotations of the metastatic areas. We first cropped these WSIs at 10x magnification into 512x512 image patches, then removed image patches with entropy less than 15 as background, and marked patches with more than 30% cancer area as positive, resulting in 186,604 image patches, of which 8,117 were marked as positive (4.3%).

TCGA-RENAL This dataset comes from The Cancer Genome Atlas (TCGA) project [17], covering high-resolution WSIs of the three main subtypes of renal cell carcinoma (RCC): clear cell renal cell carcinoma (ccRCC), papillary renal cell carcinoma (pRCC), and chromophobe renal cell carcinoma (chRCC). ccRCC is the most common subtype, characterized by clear cells containing abundant lipids and glycogen. pRCC follows, characterized by papillary structures formed on the surface of tumor cells. chRCC is relatively rare, with larger cells and granular cytoplasm. All WSIs are rigorously reviewed by professional pathologists to ensure the representativeness and quality of the data. To fully verify the effectiveness of our proposed method, we collected 910 WSIs from TCGA, and then organized experts to delineate all WSIs at the pixel level, including delineation of cancerous and non-cancerous areas. Subsequently, through preprocessing processes such as WSI segmentation and background removal similar to CAMELYON16, we obtained instance-level labels for the complete training and testing sets. Finally, all WSIs were divided into training and testing sets in a ratio of 70% and 30%, respectively.

A.2 Implementation Details

FAST employs the image encoder and text encoder from the pre-trained CLIP-RN50 [41]. The cache capacity is typically determined by the number of annotated and unannotated instances in the training set after each sampling. For bag shot greater than 4, we used a core set selection strategy to prevent excessive caching. For the CAMELYON16 and TCGA-RENAL datasets, we set the number of core sets to 1000 and 2000, respectively. We set the number of learnable tokens to 10. During training, we utilized the Adam optimizer with learning rates set as follows: 0.001 for the feature cache, 0.01 for the label cache, and 0.001 for the tokens in the prior branch. We fine-tune our model with batch size of 4096 for 20,000 epochs. All models are trained and tested on an RTX 3090 GPU with 24GB memory.

For Tip-Adapter, We conducted comparative experiments according to the settings of the optimal model in the original Tip-Adapter paper. For aspects that cannot be adapted to the few-shot WSI classification task, we used the following approach. We designed a set of text prompts specifically for pathology images, which has been proven superior in the CONCH comparison experiments we conducted. We used all annotated patches to build the cache model.

A.3 Additional experiments

(1) Annotation Efficiency To provide a more intuitive illustration of the annotation efficiency of our method, we first calculated the annotation ratio under different bag shots (1, 2, 4, 8, 16), with each bag containing 16 annotated instances. Then, we visualized the classification results of FAST under different annotation ratios and compared them with the fully supervised method. The results for TCGA-RENAL dataset are presented in Figure 5. It can be observed that the classification AUC of FAST rapidly increase with the growth of bag shots. When the bag shot reaches 8, FAST achieves results comparable to fully supervised methods. At this point, the annotation ratio of FAST on the TCGA-RENAL dataset is only 0.0067% of the total instance annotation. When the bag shot reaches 16, the annotation only accounts for 0.013% of all instances. Even under such extreme minimal annotation, FAST can still achieve results close to fully supervised methods, especially in bag classification results. For chRCC, pRCC, and ccRCC, FAST's average bag-level AUC reaches 99.64%, 96.24%, and 97.70% respectively compared to the fully supervised method. This result fully demonstrates the advantage of FAST in annotation efficiency.

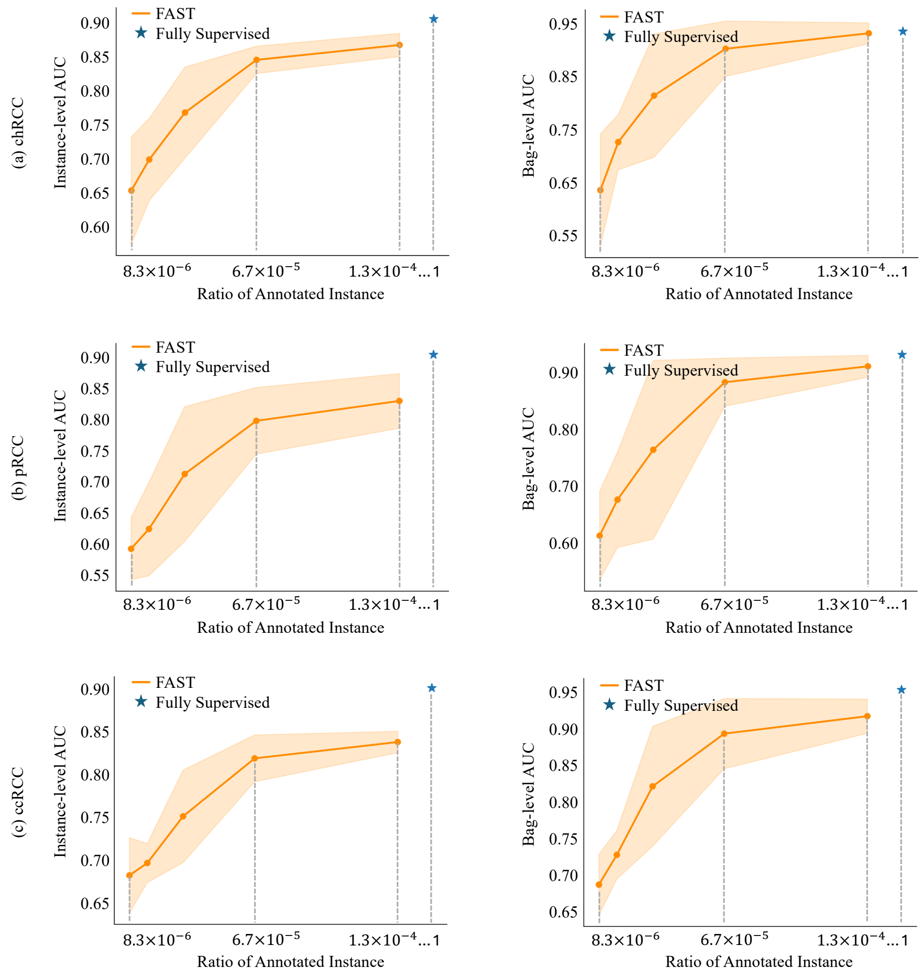

Figure 5: Results of FAST on TCGA-RENAL dataset under different annotation ratio.

(2) The Number of Annotated Instance The above experiments indicate that achieving good results only require annotating 16 instances per bag. To further investigate the influence of the number of annotated instances per bag, we conducted experiments on the CAMELYON dataset with the number of annotated instances per bag set to 4, 16, and 64, respectively. The results are shown in Figure 6. It can be observed that when the number of bags is small and there are only 1, 2, or 4 bags per class, the instance-level and bag-level classification AUCs are significantly higher when 64 instances are annotated per bag compared to only annotating 4 or 16 instances. However, as the number of bags increases to 8 and 16, the AUC of FAST gradually converge. Regardless of whether each bag is annotated with 4, 16, or 64 instances, FAST achieves similar results in instance-level AUC and bag-level AUC. These experimental results indicate that when the number of bags is extremely small, increasing the number of annotated instances can effectively improve the performance of FAST. However, when the number of bags increases to a certain level, even if only a small number of instances are annotated in each bag, FAST can achieve results similar to annotating a large number of instances. This indicates that FAST's ability to achieve higher accuracy than other methods primarily relies on its learning from unlabeled instances. It also suggests that in practical applications of FAST, if there are a large number of bags, the requirement for labeling instances can be appropriately reduced.

(3) Core Set Size Few-shot learning of WSI classification differs from conventional few-shot learning. Even with only several WSIs, they may produce tens of thousands or even hundreds of thousands of patches. Therefore, to avoid optimization difficulties caused by excessively large caches,

Table 4: Results of FAST on the CAMELYON16 dataset under different core set sizes

Bag Shot	Instance Shot	Core Set Size	Instance-level AUC	Bag-level AUC
4	16	100	0.8845±0.0525	0.7295±0.0743
		500	0.8770±0.0643	0.7210±0.1157
		1000	0.8860±0.0546	0.7554±0.0766
		2000	0.8985±0.0279	**0.7595±0.0478**
		5000	**0.9027±0.0342**	0.7499±0.0641
8	16	100	0.8827±0.0303	0.7179±0.0712
		500	0.8903±0.0162	0.7549±0.0445
		1000	0.8957±0.0132	0.7773±0.0319
		2000	0.9075±0.0122	0.7848±0.0392
		5000	**0.9101±0.0165**	**0.8025±0.0237**
16	16	100	0.9008±0.0320	0.7353±0.0624
		500	0.9046±0.0301	0.7924±0.0484
		1000	**0.9124±0.0281**	0.8078±0.0626
		2000	0.9181±0.0173	**0.8240±0.0572**
		5000	0.9096±0.0211	0.8082±0.0544

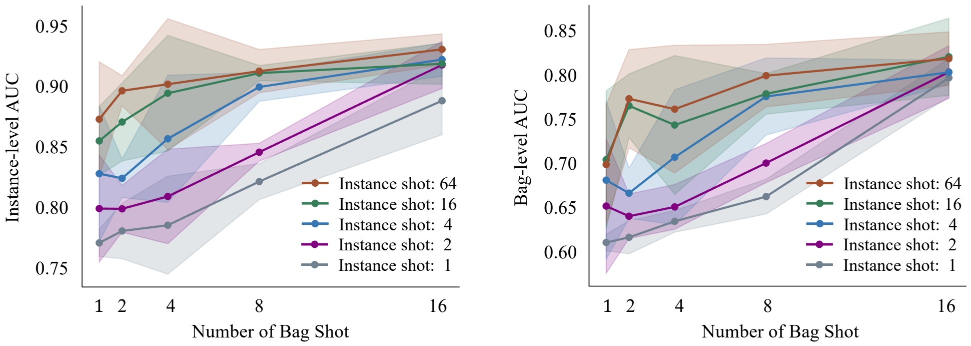

Figure 6: Results of FAST on CAMELYON16 dataset under different instance shots.

FAST employs a K-means core set selection strategy to control the size of the learnable cache. To analyze the performance of FAST under different core set sizes, we conducted experiments on the CAMELYON16 dataset with 4, 8, and 16 bags and 16 instances per bag. The results are shown in Table 4. When the core set size is only 100 or 500, FAST's bag-level AUC is significantly lower, indicating that the core set size at this time is insufficient to cover all representative samples, resulting in a loss of a large amount of information. However, when the core set size reaches 1000 or above, FAST's instance-level AUC and bag-level AUC achieve good results, indicating that FAST has good robustness to core set size. However, when the core set size is 5000 and the number of bags is 16, the results decrease instead, indicating that an overly large learnable cache is difficult to optimize. Therefore, we set the core set size to 1000 or 2000 in experiments where the number of bags is greater than 4.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the abstract and the section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Not applicable.

   Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In the abstract.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: In the abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In the section A.2 of supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In the section 4.1 and the section 4.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the section A.2 of supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research is beneficial for WSI classification.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In the CAMELYON16 paragraph of the section 4.3.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: In the section A.2 of supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: All codes and models will be made publicly accessible.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.