

# Benchmarking Pathology Foundation Models: Adaptation Strategies and Scenarios

Jaeung Lee<sup>a</sup>, Jeewoo Lim<sup>a</sup>, Keunho Byeon<sup>a</sup>, Jin Tae Kwak<sup>a,\*</sup>

<sup>a</sup>School of Electrical Engineering, Korea University, , Seoul, 02841, , Republic of Korea

## Abstract

In computational pathology, several foundation models have recently emerged and demonstrated enhanced learning capability for analyzing pathology images. However, adapting these models to various downstream tasks remains challenging, particularly when faced with datasets from different sources and acquisition conditions, as well as limited data availability. In this study, we benchmark four pathology-specific foundation models across 14 datasets and two scenarios—consistency assessment and flexibility assessment—addressing diverse adaptation scenarios and downstream tasks. In the consistency assessment scenario, involving five fine-tuning methods, we found that the parameter-efficient fine-tuning approach was both efficient and effective for adapting pathology-specific foundation models to diverse datasets within the same downstream task. In the flexibility assessment scenario under data-limited environments, utilizing five few-shot learning methods, we observed that the foundation models benefited more from the few-shot learning methods that involve modification during the testing phase only. These findings provide insights that could guide the deployment of pathology-specific foundation models in real clinical settings, potentially improving the accuracy and reliability of pathology image analysis. The code for this study is available at <https://github.com/QuIIL/BenchmarkingPathologyFoundationModels>.

*Keywords:* computational pathology, foundation model, fine-tuning, few-shot learning

## 1. Introduction

Foundation models have recently gained much attention in computational pathology for their superior capability to handle a wide range of downstream tasks [1]. In computational pathology, there are numerous downstream tasks including image classification, segmentation, and registration [2, 3]. Traditionally, we build independent and task-specific models to tackle these tasks, of which each requires sufficient high-quality data for model training and evaluation. However, this conventional approach poses substantial challenges due to the need for large, labeled datasets and the time-consuming nature of developing and fine-tuning each task-specific model. With the increase in the number of datasets and advances in model architecture and learning techniques, several pathology-specific foundation models are available [4, 5, 6, 7, 8, 9, 10]. Despite their success and potential, our understanding of these pathology-specific foundation models is limited. It remains unclear how to best utilize these models for the specific downstream tasks. Therefore, there is a demand to set up benchmarks for the pathology-specific foundation models and to gain further insights into these models.

\*Corresponding author

Developing pathology-specific foundation models generally require four steps: 1) *Model Selection*: select a base model that learns the general knowledge from pathology data; 2) *Data Preparation*: prepare large-scale pathology datasets to teach the model; 3) *Optimization*: identify a learning strategy to teach the model using the data; and 4) *Evaluation*: investigate the trained model on various use cases. Specifically, first, one needs to choose the base model among a great deal of artificial intelligence or machine learning models. Among various models, the existing foundation models often adopt Transformers as the base model for the enhanced learning ability especially from a large amount of data. Second, one needs to obtain large-scale datasets to train the chosen model. Many of the existing works utilize pathology images that are publicly available such as those from The Cancer Genome Atlas (TCGA) [11] which contains 29,000 whole slide images (WSIs) from 25 anatomic sites and covering 32 cancer subtypes with differing pathological conditions and image qualities. Some others adopt their own private pathology image datasets that are used either independently [7, 8, 9] or in combination with TCGA [4, 5]. Third, one needs to identify a suitable learning strategy to optimize the base model on the large-scale datasets. The existing models are mainly optimized or trained using self-supervised learning (SSL). SSL is a powerful technique that enables learning useful representations/knowledge from the input data with and without data annotations by utilizing the intrinsic property or structure of the data [12, 13, 14]. This is especially beneficial for pathology images due to the exceptional size of WSIs which are Giga-pixel sized and the scarcity and difficulty of data annotation. Fourth, upon the completion of the optimization, one needs to assess the performance of the foundation models. The existing works evaluate various downstream tasks across different organs and diseases by fully or partially adjusting the weights of the optimized (or pre-trained) foundation models. The number and type of downstream tasks differ one from the other and the performance varies depending on the tasks and the foundation models [7, 8, 9, 10].

Most pathology-specific foundation models are built based upon Transformers trained on public and/or private large-scale datasets using variants of SSL and evaluated on various downstream tasks via fine-tuning. Several research efforts have been made to further improve the efficiency and effectiveness of these foundation models. These, by and large, involve the advancement in architecture of the base model, the inclusion of larger and more diverse datasets, and the enhancement of SSL or other learning algorithms similar to the development trends observed in other artificial intelligence models. Along with these observations, we have noticed that the previous studies have not fully explored the impact of the adaptation strategies. Most existing works adopt linear probing or fine-tuning to adapt the foundation models to downstream tasks. Various adaptation techniques are available, but their effects on the foundation models and downstream tasks remain unknown. Moreover, the downstream tasks usually include single in-domain datasets, which is insufficient for comprehensively investigating the robustness of the foundation models. Hence, two critical questions naturally arise: 1) *are the existing models strong enough to serve as the de facto foundation models?* 2) *what is the optimal strategy to use or adapt the foundation models to downstream tasks?*

To address these two questions and to deepen our understanding of the pathology-specific foundation models, we conduct an in-depth analysis of the foundation models and their behavior under various settings and conditions. Specifically, we employ four pathology-specific foundation models including CTransPath [4], Lunit [5], Phikon [6],

and UNI [7] and adapt each of the four models to various downstream tasks with 14 datasets and 4 tasks under two experimental scenarios for consistency assessment and flexibility assessment. In the consistency assessment scenario, we evaluate how well the foundation models adapt to different datasets within the same task. This involves various fine-tuning strategies such as linear probing, full fine-tuning, partial fine-tuning, and parameter-efficient fine-tuning (PEFT) to identify the most effective approach to adjust the foundation models for carrying out downstream tasks. In the flexibility assessment scenario, we examine how well the foundation models adapt to the datasets across varying tasks or domains where we explore FSL techniques to further explore the adaptability of the foundation models. Through these two scenarios, we provide a comprehensive understanding of the practical utility of the foundation models in computational pathology and identify the best practices for their deployment in clinical settings.

Our contributions can be summarized as follows:

- We conduct a comprehensive benchmarking study of the four pathology-specific foundation models, evaluating their performance across 14 datasets from 5 organs.
- We assess the utility and capability of the foundation models through consistency and flexibility assessment scenarios, providing insights into their robustness and adaptability to various downstream tasks.
- In the consistency assessment scenario, we investigate the impact of various fine-tuning strategies, including linear probing, full fine-tuning, partial fine-tuning, and PEFT, on the foundation models and their adaptation to differing datasets within the same tasks.
- In the flexibility assessment scenario, we explore the generalization capabilities of the foundation models across three distinct adaptation scenarios, such as near-domain, middle-domain, and out-domain adaptations, using various FSL methods under data-limited conditions.

## 2. Related Works

### 2.1. Self-supervised Learning and Pathology Images

SSL is a learning paradigm that enables learning from unlabeled or partially labeled data by leveraging self-supervised signals that are generated using the intrinsic structure of data. SSL has demonstrated impressive performance in representation learning for various data types including images [15, 16, 17], videos [18, 19, 20], and text [21, 22, 23]. SSL has evolved in various ways. A majority of SSL methods are built based upon contrastive learning, which aims to improve the discriminative representation of data. For instance, SimCLR [24] and MoCo [15] applied data augmentation techniques to maximize the mutual information between different transformations of the same image, bringing similar image pairs closer in the embedding space and pushing dissimilar pairs farther apart. There are non-contrastive SSL approaches. For example, DINO [12] adopted a teacher-student framework using self-distillation, where both models continuously interact to each other to compute embeddings for two augmented views of the same image. BYOL [13] introduced a bi-directional learning mechanism for effective feature extraction without labels. Unlike DINO and BYOL, MAE [25] utilized an encoder-decoder architecture and reconstructed an original image from its duplicate image where a substantial

portion has masked out by minimizing the discrepancy between the two outputs obtained from the original image and the reconstructed image.

SSL has been widely adopted for pathology image analysis due to the ability to effectively utilize unlabeled data. Earlier works tend to focus on enhancing the representation power of a model for specific downstream tasks. For instance, IMPaSh [26] proposed a patch shuffling augmentation method, which shuffles the order of patches to address the domain adaptation problem in colon tissue subtyping tasks between different domains. It generated four different image variations and extract and utilized their features to perform contrastive learning. SD-MAE [25] utilized MAE to process the output of the encoder as a student and the output of the decoder as a teacher, and applied self-distillation between the encoder and decoder for image classification, cell segmentation, and object detection. Self-Path [27] adopted SSL using both pathology-specific (magnification prediction, magnification puzzles, and Hematoxylin channel prediction) and pathology-agnostic (rotation, flipping, real/fake prediction, and etc.) information. [28] utilized the SSL method that leverages the pathology images of multiple magnifications as a single sequence and predicts the relative order of magnifications among internal patches using multi-resolution contextual information. HIPT [29] exploited the hierarchical structure of WSI by generating and aggregating image tokens at various resolutions with the student-teacher knowledge distillation method.

In recent years, SSL are increasingly applied to large-scale pathology image datasets that are unlabeled or sparsely labeled [4, 5, 6, 7, 8, 9]. These studies typically leverage publicly available databases such as The Cancer Genome Atlas (TCGA). By training models on extensive unlabeled datasets, these approaches aim to harness the intrinsic patterns and features inherent to pathology images without the need for explicit annotations. Recent developments in the pathology-specific foundation models are generally built based upon SSL and have demonstrated the significant potential to serve as the foundation for various downstream tasks. SSL allows these foundation models to learn robust representations transferable between different pathology tasks, including tumor detection, grading, and subtyping, significantly reducing the need for extensive labeled datasets. For instance, Phikon [6] employed TCGA to learn pathological feature representations by applying the iBOT [14] framework, which masks parts of the images and reconstructs the masked portions to learn robust features. CTransPath [4] used TCGA and the Pathology AI Platform (PAIP) [30] dataset and proposed an enhanced contrastive learning method that selects positive instances with similar visual information from a memory bank. Moreover, several other studies utilize large-scale private datasets. For example, Lunit [5] adopted two datasets such as TCGA and TULIP, a private dataset with 1.3 million image patches and utilized data from various fields of view at  $20\times$  and  $40\times$  magnifications. It employed augmentation techniques specific to pathology images, including random vertical flip and RandStainNA for stain augmentation and DINO [12] for training. UNI [7] introduced Mass-100k with over 100 million images, trained the ViT-L model, and conducted 34 downstream tasks. Virchow [8] collected 1.5 million WSIs to train the ViT-H/14 that is evaluated on 26 downstream tasks. The performance of the model was evaluated on 31 different datasets. RudolfV [9] integrated data from over 15 laboratories, including 134,000 WSIs from 34,000 cases and assesses 12 downstream tasks by using ViT-L model. For model training, UNI, Virchow, and RudolfV employed DIONv2 [reference], which involves masked image modeling and self-distillation to learn

meaningful representations.

## 2.2. Pre-training and Fine-tuning

As the foundation models, trained either by SSL or other learning paradigms, are applied to downstream tasks, they usually undergo fine-tuning to adjust themselves to specific tasks or problems. Various fine-tuning techniques are available such as linear probing [31], full fine-tuning, partial fine-tuning, and PEFT. Linear probing is the simplest yet efficient adaptation method, wherein the final linear classifier is trained on the downstream tasks while the rest of the models remain frozen. The simplicity of the method allows for quick adaptation but may not fully utilize the potential of the foundation models. In contrast, full fine-tuning updates the weights across all layers of the models. This approach can offer improved performance for downstream tasks but is computationally expensive. Partial fine-tuning involves adjusting certain layers while keeping others frozen. PEFT is an emerging adaptation method, originated from natural language processing, that can adjust the model in a parameter-efficient fashion. Low-rank adaptation (LoRA) [32], for example, hypothesized that the learnable weights of a model reside on a low intrinsic dimension and proposed to update the rank decomposition of the weights during adaptation. LoRA and its variants have been successfully applied to various problems such as generating radiology texts [33], retinal Optical Coherence Tomography (OCT) segmentation [34], and multi-organ medical image segmentation [35]. Radiology-Llama2 [33] used instruction tuning and LoRA for radiology reports, generating coherent and clinically useful texts, outperforming conventional large language models (LLMs). SAMedOCT [34] adapted the Segment Anything Model (SAM) for retinal OCT scans with LoRA for efficient fine-tuning. MOELoRA [35] combined Mixture-of-Expert (MOE) models and LoRA for multi-task medical applications, addressing data imbalance and reducing computational costs.

Most foundation models have adopted linear probing for the adaptation to downstream tasks. To the best of our knowledge, PEFT has not been utilized for the adaptation of the pathology-specific foundation models.

## 2.3. Few-shot Learning

FSL is a technique to learn from a limited number of labeled examples, often only a few samples per class. The term “few-shot” emphasizes the efficiency of a model to adapt to new tasks while minimizing exposure to new data. FSL is particularly useful in scenarios where the acquisition of extensive annotated datasets is challenging and/or expensive. The key of FSL is the N-way K-shot learning setting, where a model is trained using only K labeled examples for each of N different classes. It usually employs a training strategy called episodic training, which simulates real learning tasks with limited data. In each episode, the model is exposed to a randomly selected subset of classes known as the support set, each represented by K examples. Then, the model predicts the class label of new examples, designated as a query set.

FSL falls under the learning paradigm of meta-learning, which learns how to learn and adapt to new tasks using a few examples only. Matching Networks (MatchingNet) [36] divided data in a support set and a query set and trained a model to maximize the similarity between the data in the query set and the corresponding data in the support set. Consequently, during testing, the model was able to dynamically adapt to new data (or classes) by assigning the label of the data from the support set that shows the highest

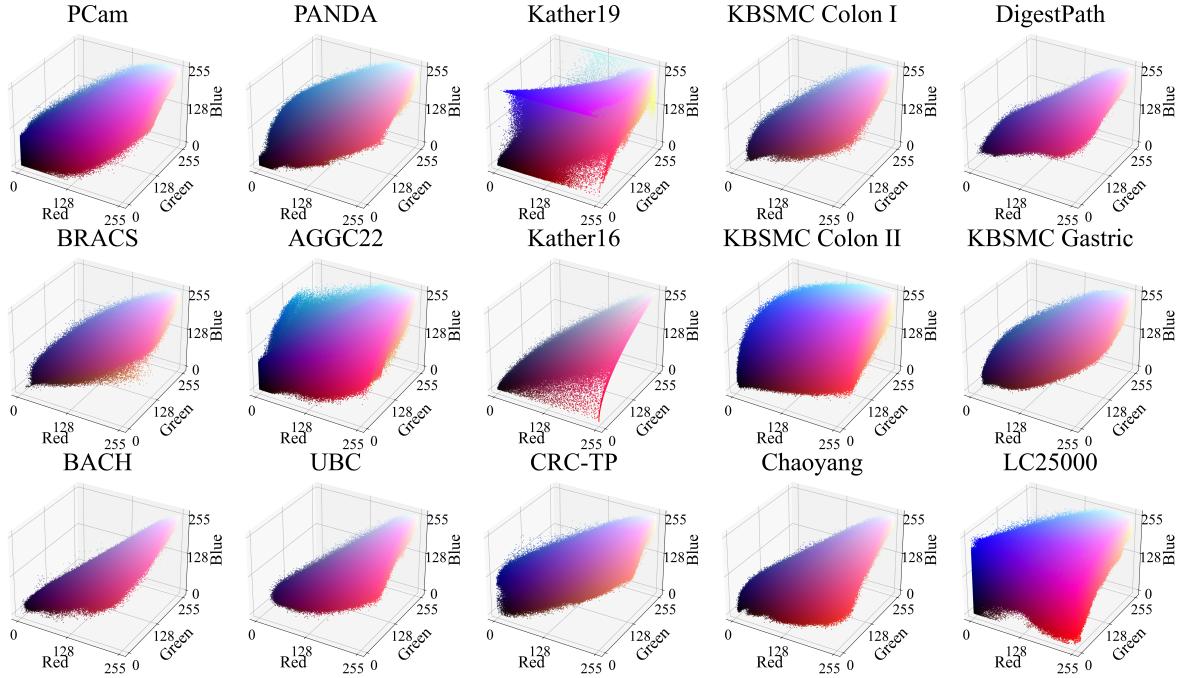


Figure 1: 3D RGB color distribution of various pathology datasets.

similarity to each test example in the query set. Prototypical Networks (ProtoNet) [37] computed the central vector (prototype) for each class based on the average of all example embeddings in the support set. To classify a new example from the query set, the model measured the Euclidean distance between the embedding of the query examples and the prototype of each class. The query examples were classified into the pertinent class with the closest prototype. Model-Agnostic Meta-Learning (MAML) [38] proposed a model to adapt quickly to new tasks through a two-step training process. In the first step, the model was trained to determine a good starting parameter set that can be quickly fine-tuned with a small number of gradient updates for various tasks. In the second step, the trained model was evaluated on new tasks, meanwhile updating the initial parameters to optimize its performance across tasks.

Several research efforts have adopted FSL for pathology image analysis. For example, [39] utilized ProtoNet to identify artifacts in pathology images. The extension of ProtoNet via k-means has been proposed to address a data distribution shift due to different scanners [40]. Furthermore, [41] investigated the impact of different FSL methods in clinical environments that deal with various cancer types and histopathological conditions. However, to the best of our knowledge, the effect of FSL on the pathology-specific foundation models has not been investigated. In this study, we adopt various FSL approaches to assess the generalization capability of the pathology-specific foundation models based upon a few examples across various tasks and datasets.

### 3. Methodology

#### 3.1. Dataset

We employ 14 publicly accessible pathology datasets to evaluate the performance of the pathology-specific foundation models on consistency and flexibility assessment

Table 1: Overview of pathology datasets with detailed characteristics.

Organ	Dataset	# Classes	# Samples	# Patients	# Patches	Patch size	Pixel size	FoV
Breast	PCam	2	400	-	327,680	96x96	0.970µm	10×
	BACH	4	-	-	14,271	512x512	0.420µm	20×
	BRACS	7	387	380	4,539	Variable size	0.250µm	40×
Colorectal	Kather19	9	86	86	100,000	224x224	0.500µm	20×
	Kather16	8	10	-	5,000	224x224	0.495µm	20×
	CRC-TP	7	20	20	196,000	150x150	-	20×
	KBSMC Colon I	4	343	343	9,857	1,024x1,024	0.247µm	40×
	KBSMC Colon II	4	45	45	110,170	1,144x1,144	0.225µm	40×
	DigestPath	2	324	-	107,982	512x512	-	20×
	ChaoYang	4	-	-	6,160	512x512	-	20×
Prostate	PANDA	4	5,158	-	735,593	512x512	0.240µm	20×
	AGGC22	5	286	-	323,697	512x512	0.500µm	20×
	UBC	4	244	-	17,066	690x690	0.250µm	40×
Gastric	KBSMC Gastric	8	98	98	206,136	1,024x1,024	0.264µm	40×
Lung & Colorectal	LC25000	5	-	-	25,000	768x768	-	-

scenarios. The datasets include pathology images with differing pathological conditions that are acquired from various organs and institutions. Due to differences in acquisition environments and digital scanners, there is substantial variability in data, such as color distribution, which can lead to domain shifts (Fig. 1). Table 1 demonstrates a summary of the entire datasets.

**Kather19** [42] comprises 100,000 image patches derived from 86 colorectal WSIs. These image patches have a spatial size of 224×224 pixels and are scanned at a pixel resolution of approximately 0.5µm under 20× magnification. This dataset includes nine distinct colorectal tissue categories: tumor tissue, simple stroma, complex stroma, immune cells, debris, normal colon mucosa, adipose tissue, mucus, and smooth muscle.

**Kather16** [43] contains 5,000 image patches, sourced from anonymized colorectal tissue slides in the pathology archive at the University Medical Center Mannheim, Heidelberg University, Germany. Each image, of size 150×150 pixels, represents one of eight different tissue types: tumor epithelium, simple stroma, complex stroma, immune cells, debris, mucosal glands, adipose tissue, and background.

**CRC-TP** [44] consists of 196,000 image patches, each of size 150×150 pixels, derived from 20 WSIs of colorectal tissues and scanned at 20× magnification from University Hospitals Coventry and Warwickshire. The dataset categorizes image patches into seven classes: tumor, inflammatory, stroma, complex stroma, necrotic, benign, and smooth muscle.

**KBSMC Colon** [45] comprises 120,123 image patches derived from colorectal WSIs and tissue microarrays (TMAs) from 45 patients. Each patch has a spatial size of 512×512 pixels with a pixel size of 0.2465µm, which is digitized at 40× magnification. The patches are categorized into four classes based on the differentiation level of cancer: benign, well-differentiated, moderately differentiated, and poorly differentiated. The dataset is divided into two parts: KBSMC Colon I and KBSMC Colon II, based on the time of acquisition and choice of digital scanners, introducing variations between the two datasets. KBSMC Colon I is digitized with an Aperio digital slide scanner (Leica Biosystems), while KBSMC Colon II is digitized with a NanoZoomer digital slide scanner (Hamamatsu Photonics

K.K.).

**Chaoyang** [46] consists of 6,160 image patches, of size  $512 \times 512$  pixels, derived from colon tissues, collected from 324 patients at Chaoyang Hospital, affiliated with Capital Medical University in Beijing. These image patches are scanned at  $20\times$  magnification and classified into four categories: normal, serrated, adenocarcinoma, and adenoma.

**DigestPath** [47] includes a colonoscopy tissue dataset for the automatic segmentation and classification of colorectal tissues. This dataset contains malignant and benign samples. The training set of malignant samples comprises 250 images with pixel-level annotations from 93 WSIs, while the benign samples consist of 410 images from 231 WSIs. All WSIs are scanned at  $20\times$  magnification using a KFBIO FK-Pro-120 slide scanner (KFBio).

**PANDA** [48] is a dataset for prostate cancer diagnosis and Gleason grading. The dataset consists of 88,199 image patches derived from 5,158 WSIs, digitized at a magnification of  $20\times$ . Each patch is  $512 \times 512$  pixels in size and includes pixel-level annotations that classify the tissue into four categories: benign, grade 3, grade 4, and grade 5.

**AGGC22** [49] is obtained from the training set of the Automated Gleason Grading Challenge 2022. This dataset includes three distinct subsets, digitized at  $20\times$  magnification. Two subsets are scanned using an Akoya Biosciences scanner, while the third subset is obtained using six different scanners: Akoya Biosciences, KFBio, Leica, Olympus, Philips, and Zeiss. It comprises 323,697 prostate image patches of size  $512 \times 512$  pixels extracted from 249 whole mount images and 37 biopsy images. The patches are categorized into five classes: stroma, benign, grade 3, grade 4, and grade 5.

**UBC** [50] is part of the training set of the Gleason2019 challenge. The dataset comprises 17,066 prostate image patches, each contains  $690 \times 690$  pixels, derived from 244 prostate tissue cores. These samples are digitized with an Aperio digital slide scanner (Leica Biosystems) at a  $40\times$  magnification and annotated by six pathologists at the Vancouver Prostate Centre. The patches are categorized into 4 classes: benign, grade 3, grade 4, and grade 5.

**PCam** [51] includes 327,680 breast image patches that are extracted from the CAMELYON16 Challenge dataset [52], which contains 400 WSIs of sentinel lymph node sections. These slides are acquired and digitized at two different centers using two different scanners: the Pannoramic 250 Flash II (3DHISTECH Ltd.) with  $20\times$  magnification and the NanoZoomer-XR Digital slide scanner C12000-01 (Hamamatsu Photonics K.K) with  $40\times$  magnification. The images are undersampled to  $10\times$  magnification. Each patch measures  $96 \times 96$  pixels. The dataset consists of two classes: tumor and normal, with an equal number of patches in each category.

**BRACS** [53] includes 547 breast WSIs collected from 189 patients, along with 4,539 regions of interests (ROIs). All slides are scanned with an Aperio AT2 (Leica Biosystems) at  $40\times$  magnification. Each ROI is annotated by consensus among three pathologists. The dataset encompasses a range of lesion types, including benign, malignant, and atypical, which are subdivided into seven classes: normal, pathological benign, usual ductal hyperplasia, flat epithelial atypia, atypical ductal hyperplasia, ductal carcinoma in situ,

and invasive carcinoma.

**BACH** [54] consists of 400 pathology image patches of breast tissue. Each patch has  $2048 \times 1536$  pixels, digitized at  $20\times$  magnification with a Leica SCN400 digital slide scanner (Leica Biosystems). These are collected for the Grand Challenge on Breast Cancer Histology held during the ICIAR 2018 conference. The dataset is divided into four classes: normal, benign, in situ carcinoma, and invasive carcinoma.

**KBSMC Gastric** [55] is derived from 98 gastric WSIs collected from 98 patients, resulting in 206,136 image patches. Each patch has a spatial size  $1,024 \times 1,024$  pixels, scanned at  $40\times$  magnification using an Aperio digital slide scanner (Leica Biosystems) with a pixel size of  $0.2635\mu\text{m}$ . The dataset has eight categories: benign, tubular well-differentiated adenocarcinoma, tubular moderately-differentiated adenocarcinoma, tubular poorly-differentiated adenocarcinoma, gastric carcinoma with lymphoid stroma, papillary carcinoma, mucinous carcinoma, and poorly cohesive carcinoma (including signet ring cell carcinoma and other poorly cohesive types).

**LC25000** [56] includes 25,000 image patches categorized into five classes with 5,000 images each, representing colon adenocarcinoma, benign colonic tissue, lung adenocarcinoma, lung squamous cell carcinoma, and benign lung tissue. All images are resized to  $768 \times 768$  pixels.

### 3.2. Foundation Models

We employ four pathology-specific foundation models including CTransPath, Lunit, Phikon, and UNI that are pre-trained on a large collection of pathology image datasets using SSL-based methods.

**CTransPath** [4] adopts a hybrid architecture combining a convolutional neural network (CNN) and a multi-scale Swin Transformer facilitating a collaborative local-global feature extraction. CTransPath is pre-trained on a large unlabeled dataset of pathology images from TCGA and PAIP, comprising approximately 15 million image patches cropped from over 30,000 WSIs by leveraging the semantically relevant contrastive learning (SRCL).

**Lunit** [5] utilizes DINO [12] to train ViT-S on 32.6 million image patches obtained from TCGA and TULIP. TULIP is a private dataset that contains 13.6 million image patches of size  $512 \times 512$  pixels. It adopts domain-specific data augmentation and field of view adjustments to optimize the model for high accuracy and reliability.

**Phikon** [6] adopts the iBOT [14] framework, which is a SSL approach that uses Masked Image Modeling (MIM), to train ViT-B on a dataset comprising over 40 million pathology images across 16 different cancer types originated from TCGA. Given an image, MIM randomly masks some regions and reconstructs the masked regions to learn useful and meaningful representations.

**UNI** [7] employs DINoV2, which is a SSL approach that uses self-distillation and MIM, to train ViT-L on a private large-scale dataset named Mass-100K. Self-distillation aligns the prediction distributions between the student and teacher networks to enhance learning stability, while MIM focuses on reconstructing masked regions of an image to

Table 2: Summary of four pathology-specific foundation models.

Model	Backbone	Training method	Dataset	# WSIs (K)	# Patches (M)	Patch size	FoV
CTransPath	CNN + Swin	SRCL	TCGA+PAIP	32	15.4	1,024×1,024	20×
Lunit	ViT-S	DINO	TCGA+TULIP	36	32.6	512×512	20×, 40×
Phikon	ViT-B	iBOT	TCGA	6	43.3	224×224	20×
UNI	ViT-L	DINOv2	MASS-100k	100	100	256×256	20×

capture meaningful representations. The Mass-100K dataset comprises over 100 million images from more than 100,000 WSIs across 20 major tissue types.

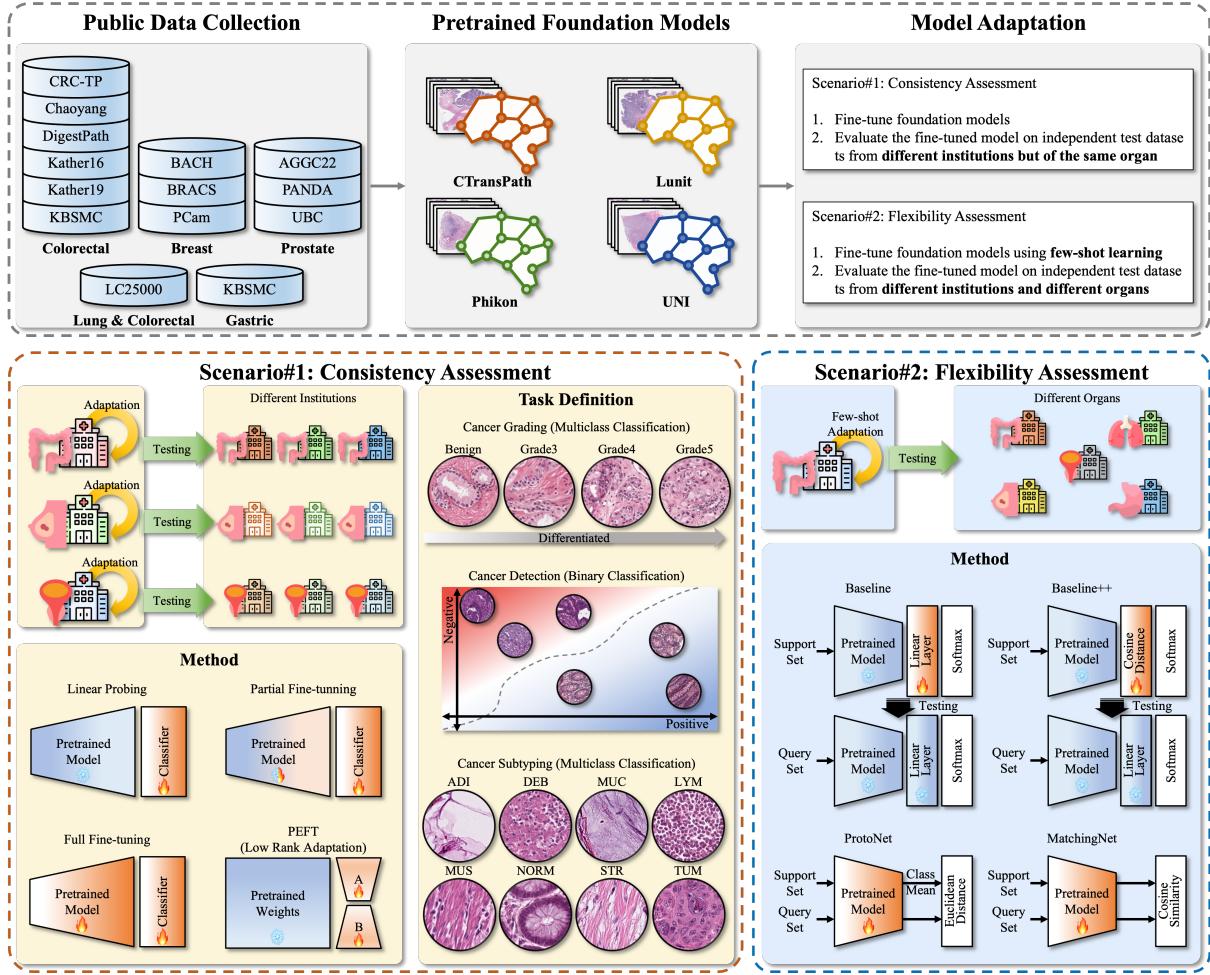


Figure 2: Overview of our benchmark study.

### 3.3. Benchmark Study Design: Consistency Assessment Scenario

The consistency assessment scenario is designed to investigate various fine-tuning methods on the pathology-specific foundation models and to identify the most suitable fine-tuning method for seamless and consistent adaptation to downstream tasks. Given a downstream task, we utilize multiple datasets acquired from various institutions and acquisition environments to assess the quality of fine-tuning methods in the context of generalization ability and domain shift issues.

### 3.3.1. Task Definition

We define four tasks with multiple datasets from three organs — breast, prostate, and colon.

**Breast Cancer Detection.** Breast cancer detection involves the binary classification of breast tissues into tumor and non-tumor categories. Three breast cancer datasets are considered: PCam, BRACS, and BACH. PCam is divided into a training subset, a validation subset, and a test subset. We use the training subset to fine-tune the foundation models and the validation subset to choose the best checkpoint of the models that is applied to the test subset and the other two datasets (BRACS and BACH). BRACS and BACH are re-grouped into two categories, i.e., tumor and non-tumor, and are designated as independent unseen datasets to assess the generalization performance of the foundation models in breast cancer detection. In BRACS, each ROI has a width of  $\leq 17,611$  pixels and a height of  $\leq 13,462$  pixels. For our experiments, we extract patches of  $256 \times 256$  pixels from these ROIs.

**Colorectal Cancer Detection.** Colorectal cancer detection is a binary classification task that distinguishes tumor colorectal tissues from and non-tumor colorectal tissues. In this task, we employ four colorectal cancer datasets: KBSMC Colon I, KBSMC Colon II, DigestPath, and Chaoyang. Following [45], we split KBSMC Colon I into a training subset, a validation subset, and a test subset for model fine-tuning and evaluation as used above. The fine-tuned foundation models are evaluated on the test subset of KBSMC Colon I and the other three datasets (KBSMC Colon II, DigestPath, and Chaoyang). We note that KBSMC Colon I and KBSMC Colon II were acquired from the same institution but were digitized at different times using different digital slide scanners.

**Colorectal Sub-Typing.** Colorectal sub-typing is a task to classify colorectal tissues into seven sub-types including tumor, inflammatory, stroma, complex stroma, necrotic, benign, and smooth muscle. We employ Kather19 for fine-tuning and use Kather16 and CRC-TP as independent unseen datasets. Kather19 was split into 70% training set, 15% validation set, and 15% testing set. Since Kather19 and Kather16 have nine and eight categories respectively, we follow [57] to re-group both datasets into seven categories such as adipose, background, debris, lymphocytes, normal, stroma, and tumor. We accomplish this by grouping stroma/muscle and debris/mucus as stroma and debris, respectively. Similarly, stroma/muscle in the CRC-TP dataset are grouped as stroma. Due to inconsistencies in the definition of complex stroma between Kather16 and CRC-TP, complex stroma is excluded from both datasets.

**Prostate Cancer Grading.** Prostate cancer grading is a task that aims to classify prostate cancer tissues into benign, grade 3, grade 4, and grade 5. For this task, we employ PANDA for fine-tuning, which is split into a training subset, a validation subset, and a test subset, following [58]. Then, we use UBC and AGGC22 for independent evaluation.

### 3.3.2. Fine-tuning Methods

We consider four fine-tuning methods, such as fully supervised learning, linear probing, full fine-tuning, partial fine-tuning, and PEFT and one training method, fully supervised learning, to adjust the pathology-specific foundation models for each downstream task.

Fully supervised learning involves initializing all weights of the foundation models and training from scratch. This method does not utilize the pre-trained weights and is used to evaluate the **Baseline** performance of the models without any pre-trained knowledge. Linear probing freezes the entire layers in the foundation models except the last classification layer, which is tailored for downstream tasks. Full fine-tuning updates the weights across all layers of the models. Partial fine-tuning adjusts the weights in the bottom 50% of the layers in the models while freezing the top 50% of the layers. For PEFT, we adopt LoRA [32], which introduces a low-rank decomposition of the weight matrices for parameter-efficient fine-tuning. Suppose that we are given a weight matrix  $W_0 \in \mathbb{R}^{d \times k}$ . We update  $W_0$  with a low-rank decomposition as follows:  $W = W_0 + \Delta W = W_0 + BA$  where  $B \in \mathbb{R}^{d \times r}$ ,  $A \in \mathbb{R}^{r \times k}$ , and the rank  $r \ll \min(d, k)$ . LoRA is adopted to adjust each self-attention layer of the foundation models.  $r$  is set to 8.

### 3.3.3. Implementation Details

We train and fine-tune all models using the Adam optimizer with default parameter values ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1.0e - 8$ ) and 256 batch size. All input images are resized to 224x224 pixels. Cross-entropy loss is adopted for supervised learning during fine-tuning. We apply color jitter, random horizontal flipping, and random resizing cropping as data augmentation techniques. All the models are implemented on the PyTorch platform and executed on a workstation with four RTX A6000 GPUs.

## 3.4. Benchmark Study Design: Flexibility Assessment Scenario

The flexibility assessment scenario evaluates generalization capability of the pathology-specific foundation models to adapt to the downstream tasks with a limited number of examples for fine-tuning, in which the traditional fine-tuning methods are not applicable. Instead of the four fine-tuning methods in the consistency assessment scenario, we explore a number of FSL methods to adjust and apply the pathology-specific foundation models to new tasks involving different organ types. Specifically, the foundation models are adjusted on a meta training dataset  $X_{Meta-train}$ , evaluated on a meta validation data  $X_{Meta-val}$  to select the best checkpoints, and then applied them to a meta test dataset  $X_{Meta-test}$ .  $X_{Meta-train}$ ,  $X_{Meta-val}$ , and  $X_{Meta-test}$  are collected from different organs and/or institutes with no overlap in their corresponding label spaces:  $Y_{Meta-train}$ ,  $Y_{Meta-val}$ , and  $Y_{Meta-test}$  ( $Y_{Meta-train} \cap Y_{Meta-val} \cap Y_{Meta-test} = \emptyset$ ). This indicates that the foundation models are asked to handle a new task with unseen classes.

For FSL-based adaptation, we utilize an  $N$ -way  $K$ -shot framework. This framework constructs a series of episodes by randomly selecting  $N$  classes from the label spaces ( $Y_{Meta-train}$ ,  $Y_{Meta-val}$ , or  $Y_{Meta-test}$ ) and  $K$  examples from each class to produced a support set. Each episode also contains a query set, which consists of  $Q$  samples from the same classes as those in the support set. The support set and query set are individually and independently formed for each of the meta datasets ( $X_{Meta-train}$ ,  $X_{Meta-val}$ , or  $X_{Meta-test}$ ). We use CRC-TP and Kather19 as  $X_{Meta-train}$  and  $X_{Meta-val}$ , respectively. Though both datasets consist of colorectal tissue samples, they were acquired from different institutions with differing tissue types. For  $X_{Meta-test}$ , we employ five datasets, including KBSMC Colon, LC25000, PANDA, KBSMC Gastric, and BACH, acquired from five distinct organs such as colon, lung, prostate, gastric, and breast. These datasets serve as the basis for three adaptation tasks, each of which introduces various sources of variability such as differences in organ type, labeling criteria, and institutions where the datasets were collected. These variations are crucial for assessing the robustness and generalization capabilities of foundation models across diverse datasets and tasks.

### 3.4.1. Task Definition

Following FHist [40], we explore the flexibility of the pathology-specific foundation models by assessing three distinct adaptation tasks: 1) near-domain adaptation, 2) middle-domain adaptation, and 3) out-domain adaptation. The details of the adaptation tasks are given by:

**Near-Domain Adaptation.** The near-domain adaptation investigates the generalizability of the foundation models to datasets sourced from the same organ but obtained from different institutions than those used in  $X_{Meta-train}$  (CRC-TP). This adaptation scenario adopts KBSMC Colon as  $X_{Meta-test}$ . Specifically, KBSMC Colon contains colorectal tissues, same as CRC-TP, but these tissues were collected and digitized at a different institution. It is noteworthy that CRC-TP is labeled for colorectal tissue sub-typing with 7 classes while KBSMC Colon is labeled for 4-class cancer grading. This adaptation scenario, therefore, assesses the ability of the foundation models to generalize across differences in labeling schemes and institutions, though the tissues were sourced from the same organ.

**Middle-Domain Adaptation.** The middle-domain adaptation evaluates the model’s ability to generalize to a combination of tissue images from both the same and different organs compared to  $X_{Meta-train}$ . For this task, LC25000 is used as  $X_{Meta-test}$ , which involves tissue sample from both the lung and colon. These samples include both benign and carcinoma tissues with 5 different categories. This scenario challenges the foundation models to address variations in labeling schemes and organ and tissue types, testing its robustness in a more complex and heterogeneous domain.

**Out-Domain Adaptation.** The out-domain adaptation challenges the foundation models by testing them on tissue images from various organs and institutions that have no overlap with  $X_{Meta-train}$ . We conduct three out-domain adaptation tasks with three different datasets: PANDA, KBSMC Gastric, and BACH, with each serving as  $X_{Meta-test}$ . These three datasets involve distinct tasks in pathology. Specifically, PANDA is a dataset for 4-class cancer grading, KBSMC Gastric is used for gastric cancer sub-typing with 8 distinct categories, and BACH is for breast cancer detection. In this adaptation scenario, we examine the ability of the foundation models to generalize across entirely new domains, accounting for variations in organ and tissue types, labeling schemes, and institutions.

We note that, in our experiments, CRC-TP is employed as  $X_{Meta-train}$ . In contrast, FHist used CRC-TP in a fully supervised setting. The baseline model is replaced by the pathology-specific foundation models in our study, allowing us to explore the effectiveness of the foundation models in adapting to new tasks involving unseen classes during training.

### 3.4.2. Few-shot Methods

To adapt the pre-trained pathology foundation models to new tasks, we consider four FSL methods: ProtoNet [37], MatchingNet [36], Baseline [59], and Baseline++ [59]. Given an episode, ProtoNet calculates the prototype (mean) for each class from the feature vectors of the support set, maps the query samples to the prototypes with the closest Euclidean distance, and utilizes this information for fine-tuning the entire model and testing. MatchingNet utilizes the cosine similarity between the support set and the query set. When an episode is given, MatchingNet fine-tunes the entire model and predicts the

query samples by weighting the support samples based on their similarity scores. Unlike the previous methods, **Baseline** and **Baseline++** fine-tune the model during the testing phase, fixing the feature extractor and repeatedly training only the final linear layer. When an episode is given during the testing phase, **Baseline** trains a new linear classifier multiple times for that specific episode. Specifically, the linear classifier is trained multiple times on the support set and then used to classify the query samples. **Baseline++**, an improved version of **Baseline**, generates class prototypes (weight vectors) by retraining a linear layer multiple times with the support set from the novel classes during the fine-tuning phase. It then predicts the query samples based on the cosine similarity between the query samples and these prototypes. Furthermore, to evaluate the effectiveness of these FSL methods, we also assess the foundation models without fine-tuning by utilizing a K-nearest neighbors (KNN) [60] approach.

### 3.4.3. Implementation Details

We apply FSL-based adaptation using ProtoNet and MatchingNet to all foundation models, utilizing an  $N$ -way  $K$ -shot framework for up to 50,000 iterations on  $X_{Meta-train}$ , with  $N$  set to 4. During the FSL-based adaptation process, we validate the foundation models every 1,000 iterations using 250 randomly sampled episodes from  $X_{Meta-val}$ , selecting the best-performing models. These selected models are tested on  $X_{Test}$  using 1,000 episodes, repeated 1,000 times. **Baseline** and **Baseline++** are performed only during the testing phase. Specifically, using the support samples from each episode of  $X_{Test}$ , we fix the feature extractor of the pre-trained foundational models and repeatedly fine-tune the final linear layer. This fine-tuning of the last layer is repeated 100 times per episode. This FSL-based adaptation procedure is repeated for different values of  $K$ , which are set to 1, 5, and 10, with  $Q$  fixed at 15. Hence, the configurations include 4-way-1-shot-15-query, 4-way-5-shot-15-query, and 4-way-10-shot-15-query. All experiments are conducted with Adam optimizer using default parameter values ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1.0e - 8$ ) and an initial learning rate of 0.05. We adapt color jittering, random horizontal flipping, and random resize cropping data augmentation techniques.

## 4. Experiments and Results

We evaluate the two scenarios used in our experiments by adopting two metrics: 1) Accuracy (Acc) and 2) Macro-average F1 (F1).

### 4.1. Consistency Assessment Scenario Experiment

Fig. 3 illustrates the results of the consistency assessment of the four pathology-specific foundation models (CTransPath, Lunit, Phikon, and UNI). In total, 4 classification tasks with 13 datasets were evaluated. Among the five adaptation methods (fully supervised learning, linear probing, full fine-tuning, partial fine-tuning, and PEFT), the strength of PEFT was prominent. It achieved the best performance in 9, 10, 9, and 8 datasets for CTransPath, Lunit, Phikon, and UNI, respectively. Among others, partial fine-tuning was shown to be superior to other three methods (fully supervised learning, linear probing, and full fine-tuning), but its performance varied depending on the datasets. In order to provide further insights into the foundation models and their performance, we ranked the four foundation models and five fine-tuning methods using a win rate heatmap, which shows the ratio of the datasets each method obtained the best performance. Fig. 4 demonstrates the win rate heatmaps for the five fine-tuning methods

and four foundation models. Similar to the observations in Fig. 3, PEFT provided the highest win rates, substantially outperforming fully supervised by 100%, linear probing by 83%, Full fine-tuning by 85%, and partial fine-tuning by 75%. This indicates that PEFT is the most suitable and consistent fine-tuning method as compared to other four fine-tuning methods. Among other methods, partial fine-tuning obtained 1.0, 0.56, 0.85, and 0.25 win rate against fully supervised learning, linear probing, full fine-tuning, and PEFT, respectively, suggesting that it is comparable to linear probing and greatly inferior to PEFT. It is also note worthy that linear probing, which is widely used for its simplicity, attained the win rates of 0.92, 0.56, 0.44, and 0.17 over fully supervised learning, full fine-tuning, partial fine-tuning, and PEFT, respectively. These results, particularly the poor performance compared to PEFT, raise questions about the utility of linear probing for the adaptation purposes, since the performance is a primary concern for clinical usage. Moreover, among the four pathology-specific foundation models, the performance of UNI was striking. It had the highest win rates of 0.79, 0.79, and 0.64 against CTransPath, Lunit, and Phikon, respectively, indicating that UNI is the best performing foundation model within the four tasks and 13 datasets used in this study. Lunit was generally shown to be inferior to other three models. Phikon was slightly better than CTransPath.

#### 4.1.1. Breast Cancer Detection

Table 3 shows the results of breast cancer detection using different fine-tuning methods with the four pathology-specific foundation models. Among the five fine-tuning methods, PEFT proved to be the most beneficial for the four pathology-specific foundation models (CTransPath, Lunit, Phikon, and UNI). On PCam (fine-tuning dataset), the adoption of PEFT was able to improve Acc and F1 by 1.88~7.25% and 0.019~0.074 for CTransPath, 1.04~11.18% and 0.010~0.114 for Lunit, 0.72~14.36% and 0.007~0.148 for Phikon, and 1.94~12.14% and 0.019~0.123 for UNI, respectively, compared to other four fine-tuning methods. For the other two datasets (BACH and BRACS), we made similar observations. On BACH, PEFT consistently increased Acc and F1 by 2.14~13.61% and 0.024~0.133 for CTransPath, 0.48~23.76% and 0.007~0.237 for Lunit, 1.44~26.01% and 0.020~0.275 for Phikon, and 0.82~37.27% and 0.009~0.405 for UNI, respectively. On BRACS, PEFT also outperformed other four fine-tuning methods regardless of the foundation models except F1 by CTransPath, providing the highest F1 of 0.664.

#### 4.1.2. Colorectal Cancer Sub-Typing

In colorectal cancer sub-typing, the advantages of PEFT was obvious regardless of the datasets and foundation models (Table 5). On Kather19 (fine-tuning dataset), the combination of PEFT with the found pathology-specific foundation models (CTransPath, Lunit, Phikon, and UNI) achieved the best performance, with Acc of 98.72%, 97.76%, 97.81%, and 99.33% and F1 of 0.988, 0.978, 0.980, and 0.994, respectively. These were substantially superior to those obtained by the other four fine-tuning methods. On Kather16 and CRC-TP, the adoption of PEFT provided similar and consistent improvements. Equipped with PEFT, Acc was improved by 0.82~8.89% for CTransPath, 1.53~13.50% for Lunit, 0.02~11.86% for Phikon, and 1.30~12.37% for UNI. As for F1, PEFT outperformed other four fine-tuning methods except partial fine-tuning with Phikon, which obtained 0.831 F1. As the five fine-tuning methods are applied to CRC-TP, the strength of PEFT was remarkable, providing substantail improvements by 1.03~38.51% and 0.014~0.473 for CTransPath, 2.29~32.14% and 0.026~0.383 for Lunit, 0.91~37.25% and 0.013~0.427 for Phikon, and 2.29~38.09% and 0.010~0.509 for UNI, respectively.

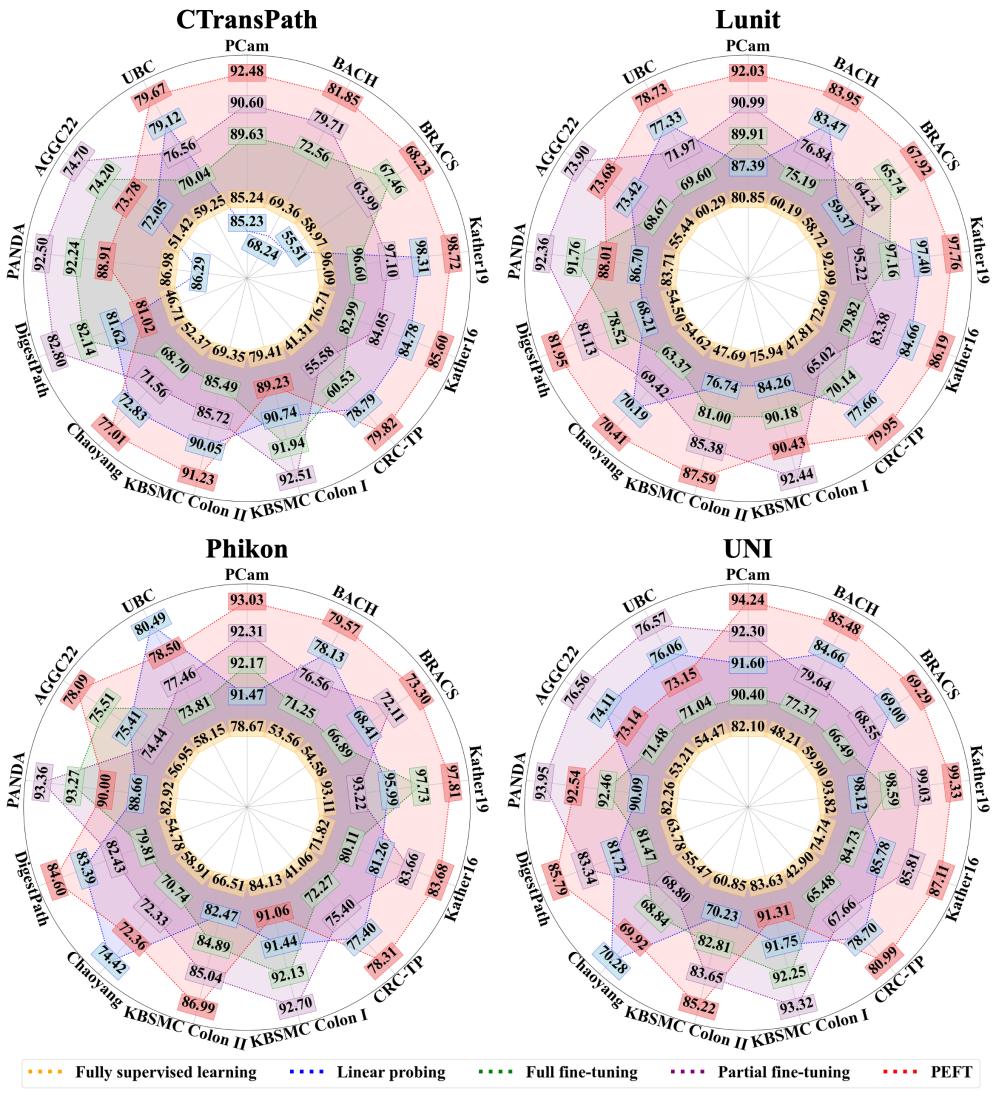


Figure 3: Results of consistency assessment scenario.

#### 4.1.3. Colorectal Cancer Detection

Table 4 demonstrates the results of colorectal cancer detection using the four foundation models and five fine-tuning methods. The impact of the five fine-tuning methods varied disproportionately depending on the datasets. On KBSCM Colon I, the application of partial fine-tuning attained the highest Acc and F1 regardless of the foundation models, outperforming that of PEFT by 1.64%~3.28% Acc and 0.016~0.039 F1. However, partial fine-tuning was not generally effective for other datasets. On KBMSC Colon II, PEFT was superior to partial fine-tuning and other fine-tuning methods, consistently enhancing both Acc and F1 across different foundation models such as 1.18~21.88% Acc and 0.010~0.264 F1 for CTransPath, 2.21~39.9% Acc and 0.026~0.391 F1 for Lunit, 1.95~20.48% Acc and 0.018~0.276 F1 for Phikon, and 1.57~24.37% Acc and 0.015~0.239 F1 for UNI, respectively. On DigestPath, similar trends were observed. The adoption of PEFT provided a consistent performance gain in F1 by 0.006~0.324, 0.012~0.304, 0.029~0.260, and 0.022~0.289 for CTransPath, Lunit, Phikon, and UNI, respectively. With respect to Acc, PEFT with Lunit, Phikon, and UNI obtained the best performance of 81.95%, 84.60%, and 85.79% but PEFT with CTransPath was inferior to Linear prob-

ing, full fine-tuning, and partial fine-tuning. On Chaoyang, PEFT proved particularly effective for CTransPath and Lunit, achieving the highest Acc of 77.01% and 70.41% and F1 of 0.768 and 0.704, respectively. However, for Phikon and UNI, PEFT was inferior to linear probing, which obtained Acc of 74.42% and 70.28% and F1 of 0.743 and 0.701, respectively.

#### 4.1.4. Prostate Cancer Grading

The results of prostate cancer grading on three test datasets are available in Table 6. On PANDA (fine-tuning dataset), partial fine-tuning showed the highest performance across different foundation models, achieving 92.36%~92.50% Acc and 0.898~0.918 F1. On the two out-of-domain datasets (AGGC22 and UBC), the results varied depending on the specific combination of fine-tuning methods and foundation models and the choice of evaluation metrics. On AGGC22, the usage of partial fine-tuning resulted in the highest Acc of 74.70%, 73.90%, and 76.56% for CTransPath, Lunit, and UNI, respectively. However, these obtained substantially poorer F1 compared to those obtained using PEFT. In terms of F1, linear probing with Lunit and UNI achieved the highest scores of 0.608 and 0.607, respectively, and PEFT with CTransPath and Phikon produced the best scores, reaching 0.602 and 0.629, respectively. For Phikon, PEFT obtained the highest Acc and F1 of 78.09% and 0.629, respectively. Notably, no other combination of fine-tuning methods and foundation models was able to attain the highest Acc and F1 simultaneously. On UBC, partial-fine tuning was inferior to either linear probing or PEFT except Acc for UNI. Linear probing managed to achieve the highest F1 of 0.616 and 0.621 for CTransPath and UNI, respectively, and the highest Acc of 80.49% for Phikon. PEFT produced the best results for CTransPath and Phikon, with the top Acc and F1 of 79.67% 0.622, respectively. It is remarkable that, for Lunit, PEFT achieved both the highest Acc (78.73%) and F1 (0.625), making it the only combination to attain the best results for both evaluation metrics at the same time.

#### 4.1.5. Computational Complexity for Consistency Assessment Scenario

We measured and compared the computational complexity of different fine-tuning methods across four pathology-specific foundation models. For each combination, we computed the number of parameters and measured the average execution time and the peak memory usage during both training and testing per image and batch. We set the batch size to 64. The results are available in Table 7. Among the four foundation models, UNI has the highest number of parameters > 303.0 million, while CTransPath and Lunit require a relatively smaller number of parameters < 303.0 million. For these foundation models, usage of linear probing, partial fine-tuning, and full fine-tuning did not add any additional parameters, whereas PEFT increased the number of parameters by approximately 1%. However, the memory requirement varied significantly depending on the fine-tuning methods, particularly during training. For instance, linear probing consumed the least memory, as it primarily adjusts only the final layer without modifying additional layers. In contrast, full fine-tuning modifies all layers, leading to the highest memory consumption, which is 3 to 13 times greater than linear probing, depending on the foundation models. During testing, the memory usage decreased as all parameters remained fixed, resulting in comparable memory requirements across different fine-tuning methods. PEFT caused only a negligible increase in memory usage due to the increase in the number of parameters. With respect to the execution time, larger models generally

Table 3: Fine-tuning adaptation results on breast cancer detection.

Test Dataset	Model	Fully supervised learning		Linear probing		Full fine-tuning		Partial fine-tuning		PEFT	
		Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
PCam	CTransPath	85.24	0.851	85.23	0.852	89.63	0.896	90.60	0.906	<b>92.48</b>	<b>0.925</b>
	Lunit	80.85	0.806	87.39	0.874	89.91	0.899	90.99	0.910	<b>92.03</b>	<b>0.920</b>
	Phikon	78.67	0.782	91.47	0.914	92.17	0.921	92.31	0.923	<b>93.03</b>	<b>0.930</b>
	UNI	82.10	0.819	91.60	0.916	90.40	0.904	92.30	0.923	<b>94.24</b>	<b>0.942</b>
BACH	CTransPath	69.36	0.679	68.24	0.677	72.56	0.684	79.71	0.786	<b>81.85</b>	<b>0.810</b>
	Lunit	60.19	0.593	83.47	0.823	75.19	0.731	76.84	0.751	<b>83.95</b>	<b>0.830</b>
	Phikon	53.56	0.508	78.13	0.763	71.25	0.661	76.56	0.738	<b>79.57</b>	<b>0.783</b>
	UNI	48.21	0.444	84.66	0.840	77.37	0.765	79.64	0.787	<b>85.48</b>	<b>0.849</b>
BRACS	CTransPath	58.97	0.579	55.51	0.490	67.46	<b>0.664</b>	63.99	0.607	<b>68.23</b>	0.647
	Lunit	58.72	0.568	59.37	0.494	65.74	0.637	64.24	0.596	<b>67.92</b>	<b>0.645</b>
	Phikon	54.58	0.533	68.41	0.651	66.89	0.654	72.11	0.708	<b>73.30</b>	<b>0.720</b>
	UNI	59.90	0.593	69.00	0.651	66.49	0.646	68.55	0.653	<b>69.29</b>	<b>0.662</b>

Table 4: Fine-tuning adaptation results on colorectal cancer sub-typing.

Test Dataset	Model	Fully supervised learning		Linear probing		Full fine-tuning		Partial fine-tuning		PEFT	
		Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
Kahter19	CTransPath	96.09	0.960	98.31	0.983	96.60	0.969	97.10	0.973	<b>98.72</b>	<b>0.988</b>
	Lunit	92.99	0.931	97.40	0.975	97.16	0.973	95.22	0.957	<b>97.76</b>	<b>0.978</b>
	Phikon	93.11	0.933	95.99	0.965	97.73	0.979	93.22	0.939	<b>97.81</b>	<b>0.980</b>
	UNI	93.82	0.936	98.12	0.983	98.59	0.986	99.03	0.991	<b>99.33</b>	<b>0.994</b>
Kahter16	CTransPath	76.71	0.756	84.78	0.849	82.99	0.823	84.05	0.838	<b>85.60</b>	<b>0.855</b>
	Lunit	72.69	0.716	84.66	0.843	79.82	0.788	83.38	0.831	<b>86.19</b>	<b>0.863</b>
	Phikon	71.82	0.707	81.26	0.802	80.11	0.792	83.66	<b>0.831</b>	<b>83.68</b>	0.828
	UNI	74.74	0.731	85.78	0.854	84.73	0.843	85.81	0.855	<b>87.11</b>	<b>0.869</b>
CRC-TP	CTransPath	41.31	0.249	78.79	0.708	60.53	0.485	55.58	0.378	<b>79.82</b>	<b>0.722</b>
	Lunit	47.81	0.328	77.66	0.685	70.14	0.538	65.02	0.509	<b>79.95</b>	<b>0.711</b>
	Phikon	41.06	0.278	77.40	0.692	72.27	0.604	75.40	0.652	<b>78.31</b>	<b>0.705</b>
	UNI	42.90	0.219	78.70	0.718	65.48	0.522	67.66	0.557	<b>80.99</b>	<b>0.728</b>

required more time to execute, especially during training. Among the fine-tuning methods, full fine-tuning was typically the slowest during training, while linear probing was the fastest. However, during testing, the execution time across the fine-tuning methods was more or less the same within each foundation model.

#### 4.2. Flexibility Assessment Scenario Experiment

Fig. 5 demonstrates the results of flexibility assessment over five FSL methods (KNN, MatchingNet, ProtoNet, **Baseline**, and **Baseline++**) with shot counts of 1, 5, and 10. Three adaptation tasks, including near-domain, middle-domain, and out-domain adaptation tasks, were evaluated. The out-domain adaptation task contains three sub-tasks such as prostate cancer grading, gastric cancer sub-typing, and breast tissue sub-typing. Within the five FSL methods, **Baseline** and **Baseline++** generally stood out as the top-performing methods. MatchingNet and ProtoNet were found to be substantially less effective in the adaptation tasks compared to other methods. KNN was comparable to **Baseline** and **Baseline++**, particularly in the 1-shot scenario.

Moreover, we recorded the ranking of the five FSL methods over the three adaptation tasks and computed the win rates. Fig. 6 shows the win rate heatmaps for the five FSL methods in the 1-shot, 5-shot, and 10-shot scenarios. In the 1-shot scenario, **Baseline** produced the highest win rates, obtaining 0.65 against KNN, 1.00 against both MatchingNet and ProtoNet, and 0.65 against **Baseline++**. Both **Baseline++** and KNN were also dominant compared to MatchingNet and ProtoNet, each achieving win rates of 1.00 against these two models. **Baseline++** and KNN were comparable to each other. Re-

Table 5: Fine-tuning adaptation results on colorectal cancer detection.

Test Dataset	Model	Fully supervised learning		Linear probing		Full fine-tuning		Partial fine-tuning		PEFT	
		Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
KBSMC Colon I	CTransPath	79.41	0.776	90.74	0.902	91.94	0.917	<b>92.51</b>	<b>0.922</b>	89.23	0.883
	Lunit	75.94	0.736	84.26	0.835	90.18	0.898	<b>92.44</b>	<b>0.921</b>	90.43	0.901
	Phikon	84.13	0.832	91.44	0.911	92.13	0.919	<b>92.70</b>	<b>0.923</b>	91.06	0.907
	UNI	83.63	0.833	91.75	0.913	92.25	0.920	<b>93.32</b>	<b>0.930</b>	91.31	0.910
KBSMC Colon II	CTransPath	69.35	0.636	90.05	0.890	85.49	0.846	85.72	0.849	<b>91.23</b>	<b>0.900</b>
	Lunit	47.69	0.476	76.74	0.736	81.00	0.803	85.38	0.841	<b>87.59</b>	<b>0.867</b>
	Phikon	66.51	0.584	82.47	0.818	84.89	0.840	85.04	0.842	<b>86.99</b>	<b>0.860</b>
	UNI	60.85	0.603	70.23	0.700	82.81	0.819	83.65	0.827	<b>85.22</b>	<b>0.842</b>
Chaoyang	CTransPath	52.37	0.524	72.83	0.727	68.70	0.687	71.56	0.716	<b>77.01</b>	<b>0.768</b>
	Lunit	54.62	0.546	70.19	0.684	63.37	0.633	69.42	0.694	<b>70.41</b>	<b>0.704</b>
	Phikon	58.91	0.581	<b>74.42</b>	<b>0.743</b>	70.74	0.707	72.33	0.723	72.36	0.722
	UNI	55.47	0.554	<b>70.28</b>	<b>0.701</b>	68.84	0.688	68.80	0.688	69.92	0.699
DigestPath	CTransPath	46.71	0.448	81.62	0.766	82.14	0.743	<b>82.80</b>	0.761	81.02	<b>0.772</b>
	Lunit	54.50	0.449	68.21	0.648	78.52	0.694	81.13	0.741	<b>81.95</b>	<b>0.753</b>
	Phikon	54.78	0.516	83.39	0.747	79.81	0.705	82.43	0.734	<b>84.60</b>	<b>0.776</b>
	UNI	63.78	0.510	81.72	0.752	81.47	0.739	83.34	0.777	<b>85.79</b>	<b>0.799</b>

Table 6: Fine-tuning adaptation results on prostate cancer grading.

Test Dataset	Model	Fully supervised learning		Linear probing		Full fine-tuning		Partial fine-tuning		PEFT	
		Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
PANDA	CTransPath	86.98	0.828	86.29	0.817	92.24	0.895	<b>92.50</b>	<b>0.898</b>	88.91	0.845
	Lunit	83.71	0.797	86.70	0.829	91.76	0.890	<b>92.36</b>	<b>0.898</b>	88.01	0.843
	Phikon	82.92	0.785	88.66	0.850	93.27	0.911	<b>93.36</b>	<b>0.911</b>	90.00	0.866
	UNI	82.26	0.775	90.09	0.868	92.46	0.899	<b>93.95</b>	<b>0.918</b>	92.54	0.902
AGGC22	CTransPath	51.42	0.361	72.05	0.569	74.20	0.592	<b>74.70</b>	0.552	73.78	<b>0.602</b>
	Lunit	55.44	0.317	73.42	<b>0.608</b>	68.67	0.494	<b>73.90</b>	0.547	73.68	0.600
	Phikon	56.95	0.349	75.41	0.575	75.51	0.579	74.44	0.572	<b>78.09</b>	<b>0.629</b>
	UNI	53.21	0.346	74.11	<b>0.607</b>	71.48	0.519	<b>76.56</b>	0.585	73.14	0.600
UBC	CTransPath	59.25	0.432	79.12	<b>0.616</b>	70.04	0.524	76.56	0.583	<b>79.67</b>	0.605
	Lunit	60.29	0.412	77.33	0.614	69.60	0.504	71.97	0.543	<b>78.73</b>	<b>0.625</b>
	Phikon	58.15	0.433	<b>80.49</b>	0.618	73.81	0.569	77.46	0.597	78.50	<b>0.622</b>
	UNI	54.47	0.426	76.06	<b>0.621</b>	71.04	0.504	<b>76.57</b>	0.568	73.15	0.606

Table 7: Computational complexity of foundation models and fine-tuning methods.

Model	Method	# Params (M)	Memory (MB/batch)		Memory (MB/image)		Time (ms/batch)		Time (ms/image)	
			Train	Test	Train	Test	Train	Test	Train	Test
CTrasnPath	Linear probing	27.523	2,363	2,355	733	595	82.59	81.55	18.83	15.02
	Partial fine-tuning	27.523	2,961	2,355	973	595	128.76	81.55	34.84	14.97
	Full fine-tuning	27.523	7,949	2,355	1,083	595	225.74	81.50	48.04	15.03
	PEFT	27.804	6,017	2,375	777	617	197.18	81.40	40.82	15.10
Lunit	Linear probing	21.667	885	879	451	443	53.39	54.41	12.71	10.49
	Partial fine-tuning	21.667	3,341	879	687	443	109.38	54.27	24.15	10.78
	Full fine-tuning	21.667	5,243	879	847	443	150.16	54.29	35.00	10.70
	PEFT	21.814	4,115	884	545	456	143.02	54.30	28.70	10.53
Phikon	Linear probing	85.802	1,549	1,541	745	717	168.64	170.71	13.88	11.93
	Partial fine-tuning	85.802	7,555	1,541	1,729	717	354.58	171.10	36.16	11.81
	Full fine-tuning	85.802	10,729	1,541	2,263	717	449.72	171.07	42.42	11.98
	PEFT	86.097	7,857	1,552	823	726	357.55	175.39	37.17	11.96
UNI	Linear probing	303.355	2,331	2,323	1,545	1,517	523.36	526.08	17.66	15.56
	Partial fine-tuning	303.355	21,555	2,323	4,451	1,517	1265.44	526.92	61.21	15.40
	Full fine-tuning	303.355	27,039	2,323	6,483	1,517	1424.93	526.30	96.91	15.70
	PEFT	303.549	15,561	2,332	2,061	1,533	1383.58	525.72	66.73	15.72

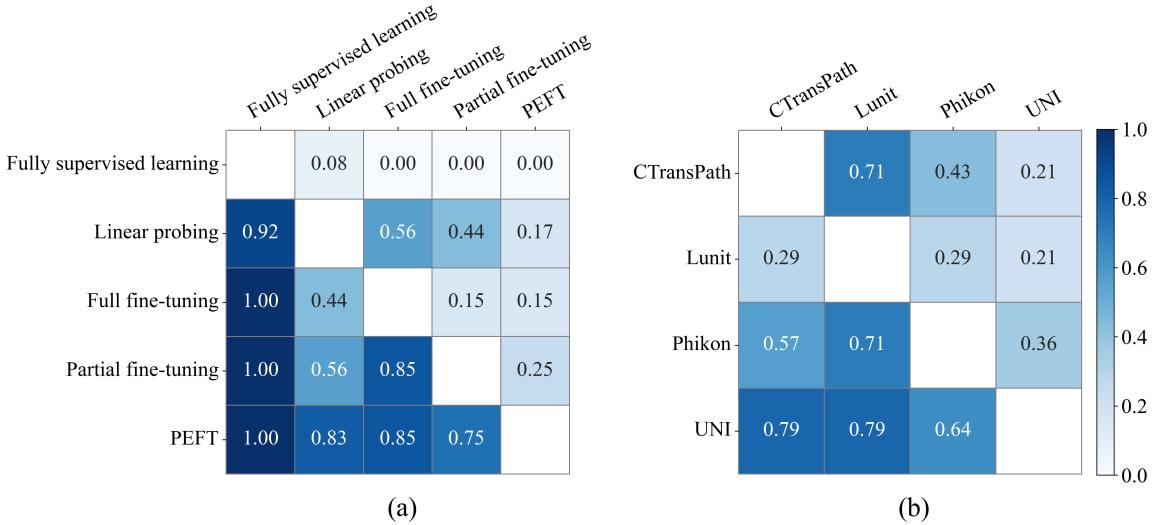


Figure 4: Win rate heatmap of (a) five fine-tuning methods and (b) 4 foundation models across 13 datasets.

garding the 5-shot and 10-shot scenarios, the dominance of **Baseline** and **Baseline++** was further highlighted with the win rates of 0.95, 1.00, and 1.00 over KNN, MatchingNet, and ProtoNet. **Baseline++** was slightly better than **Baseline** by the win rate of 0.55. Though KNN surpassed MatchingNet and ProtoNet, it was clearly inferior to both **Baseline** and **Baseline++**.

#### 4.2.1. Near-Domain Adaptation

The results of the near-domain adaptation are available in Table 8. **Baseline** generally achieved the best results for both Acc and F1 regardless of the foundation models, evaluation metrics, and number of shots, with the exception of Phikon with 10 shots and UNI with 5 and 10 shots. For these three cases, **Baseline** was ranked as the second-best model. As for other cases, **Baseline++** was typically found to be the second-best model except CTransPath with 1 shot. In a head-to-head comparison among the four pathology-specific foundation models, Phikon with **Baseline** achieved the highest Acc and F1 across different number of shots, substantially outperforming other three models by 8.37%~9.13% Acc and 0.090~0.100 F1 for 1 shot, 3.05%~4.41% Acc and 0.032~0.040 F1 for 5 shots, and 0.80%~3.20% Acc and 0.007~0.030 F1 for 10 shots. In terms of shot counts, Phikon with **Baseline** produced the best results for 1 (73.56% Acc and 0.727 F1) and 5 shots (84.57% Acc and 0.845 F1). However, for 10 shots, Phikon with **Baseline++** attained the highest Acc and F1 of 87.21% and 0.871, respectively.

#### 4.2.2. Middle-Domain Adaptation

Table 8 shows the results of the middle-domain adaptation. For CtransPath, Phikon and UNI, **Baseline++** was generally superior to other adaptation methods across different shot counts and evaluation metrics except for CTransPath with 1 shot. As for Lunit, while KNN outperformed others using 1 shot, **Baseline** produced the top performance using 5 shots (96.63% Acc and 0.966 F1) and 10 shots (96.08% Acc and 0.960 F1). With respect to the shot counts and foundation models, equipped with KNN, CTransPath achieved the best Acc of 92.15% and F1 of 0.919 with 1 shot, greatly outperforming other foundation models by 3.79%~6.09% Acc and 0.045~0.070 F1 as compared to the best results obtained by each model with 1 shot. Using 5 and 10 shots, UNI with **Baseline++**

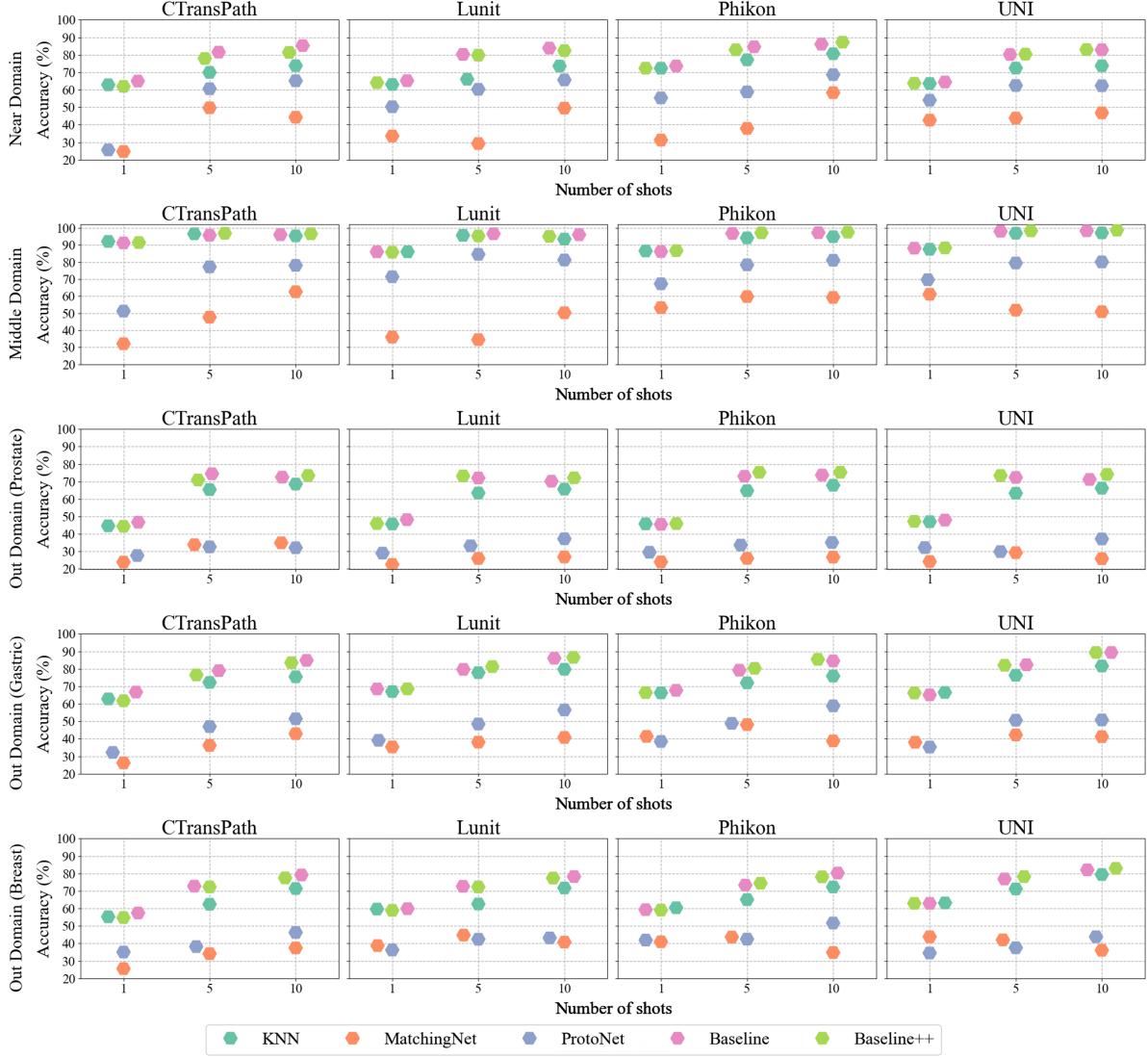


Figure 5: Results of flexibility assessment scenario.

attained the top results, with 98.37% Acc and 0.984 F1 for 5 shots and 98.76% Acc and 0.988 F1 for 10 shots. This presents an improvement of 1.19%~1.74% Acc and 0.012~0.018 F1 for 5 shots and an improvement of 1.15%~2.68% Acc and 0.012~0.028 F1 for 10 shots over the best-performing combinations from other foundation models.

#### 4.2.3. Out-Domain Adaptation: Prostate Cancer Grading

In the out-domain adaptation to prostate cancer grading (Table 9), **Baseline** and **Baseline++** were shown to be the top-performing models across various shot counts and foundation models. Specifically, for CTransPath, **Baseline** obtained the highest results for the 1 shot (46.74% Acc and 0.439 F1) and 5 shots (74.44% Acc and 0.737 F1), meanwhile **Baseline++** achieved the highest Acc of 73.46% and F1 of 0.728. As for other foundation models (Phikon, Lunit, and UNI), **Baseline++** outperformed other adaptation methods, particularly effective in scenarios with 5 shots and 10 shots. In the scenario with 1 shot, **Baseline** was beneficial for Lunit and UNI, surpassing other methods. As for Phikon, **Baseline++** obtained the highest Acc of 45.94% and the second-highest F1 of 0.427, with KNN achieving the highest F1 of 0.435.

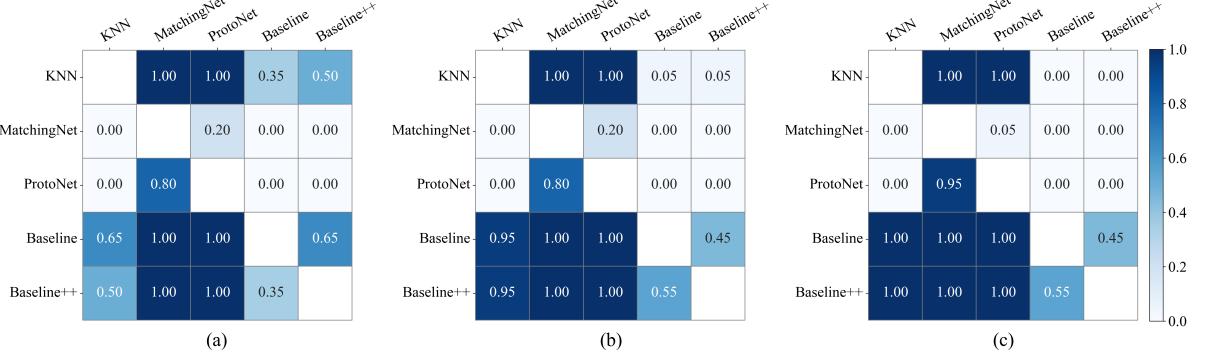


Figure 6: Win rate heatmap of (a) 1-shot, (b) 5-shot, and (c) 10-shot few-shot learning methods across 5 datasets.

#### 4.2.4. Out-Domain Adaptation: Gastric Cancer Sub-typing

Similarly, **Baseline** and **Baseline++** demonstrated effectiveness in the out-domain adaptation for gastric cancer sub-typing. (Table 9). For CTransPath, **Baseline** was consistently superior to other methods regardless of shot counts and evaluation metrics. In contrast, for Lunit, **Baseline++** delivered the best results for every shot count and evaluation metric. With Phikon, **Baseline** proved to be the most advantageous for the 1 shot scenario, and, in the scenarios with 5 and 10 shots, **Baseline++** outperformed other methods. When it comes to UNI, KNN with 1 shot obtained the best performance, while **Baseline++** achieved the highest Acc and F1 for both 5 and 10 shots. Regarding the combinations of the foundation models and shot counts, Lunit with **Baseline++** attained the best results for the 1-shot scenario, with 68.52% Acc and 0.680 F1. For the scenarios with 5 and 10 shots, the combination of UNI and **Baseline** obtained the highest scores, with 82.32% Acc and 0.824 F1 for 5 shots and 89.34% Acc and 0.894 F1 for 10 shots.

#### 4.2.5. Out-Domain Adaptation: Breast Cancer Detection

The results of the out-domain adaptation for breast cancer detection further confirmed the strength of **Baseline** and **Baseline++**, as shown in Table 9. For both CTransPath and Lunit, **Baseline** constantly proved to be the best adaptation method across different shot counts and evaluation metrics. The second-best methods varied; KNN ranked second for the 1 shot scenario while **Baseline++** took the second place in the 5 and 10 shot scenarios. With regard to Phikon and UNI, the results varied depending on shot counts and foundation models. For both Phikon and UNI, KNN was the top performer in the 1-shot scenario and **Baseline++** was the best-performing method in the 5-shot scenario. With 10 shots, **Baseline** and **Baseline++** obtained the superior performance for Phikon and UNI, respectively. Comparing the four pathology-specific foundation models, UNI proved to be the most effective adaptation method. Specifically, using 1 shot, UNI with KNN delivered the best results of 63.16% Acc and 0.622 F1. For 5 and 10 shots, UNI, paired with **Baseline++**, found to be the top-performing model, achieving an Acc of 78.22% and F1 of 0.774 F1 for 5 shots and an Acc of 82.94% and F1 of 0.830 for 10 shots.

Table 8: Few-shot learning results on near-domain and middle-domain adaptation tasks.

		Near-domain (Colorectal cancer grading)						Middle-domain (Lung and colon sub-typing)					
		1-shot		5-shot		10-shot		1-shot		5-shot		10-shot	
Model	Method	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
CTransPath	KNN	62.85	0.618	69.93	0.684	73.77	0.726	<b>92.15</b>	<b>0.919</b>	96.58	0.965	95.42	0.954
	MatchingNet	24.69	0.111	49.69	0.433	44.27	0.388	32.12	0.232	47.71	0.407	62.62	0.597
	ProtoNet	25.60	0.231	60.58	0.586	65.13	0.644	51.27	0.483	77.22	0.771	78.08	0.778
	Baseline	<b>64.96</b>	<b>0.632</b>	<b>81.52</b>	<b>0.813</b>	<b>85.25</b>	<b>0.853</b>	91.27	0.909	95.77	0.958	96.09	0.961
	Baseline++	61.93	0.608	77.88	0.775	81.34	0.814	91.50	0.911	<b>96.93</b>	<b>0.969</b>	<b>96.59</b>	<b>0.966</b>
Lunit	KNN	63.00	0.614	65.99	0.636	73.56	0.728	<b>86.06</b>	<b>0.849</b>	95.71	0.956	93.52	0.934
	MatchingNet	33.56	0.304	29.26	0.233	49.51	0.392	36.01	0.349	34.51	0.296	50.29	0.416
	ProtoNet	50.27	0.480	60.25	0.592	65.62	0.647	71.42	0.701	84.57	0.844	81.27	0.812
	Baseline	<b>65.19</b>	<b>0.636</b>	<b>80.22</b>	<b>0.800</b>	<b>83.82</b>	<b>0.838</b>	86.05	0.848	<b>96.63</b>	<b>0.966</b>	<b>96.08</b>	<b>0.960</b>
	Baseline++	64.05	0.624	79.78	0.794	82.37	0.823	85.84	0.846	95.37	0.953	95.13	0.951
Phikon	KNN	72.30	0.716	77.12	0.757	80.59	0.792	86.57	0.857	94.22	0.941	94.92	0.949
	MatchingNet	31.27	0.292	37.93	0.335	58.34	0.532	53.27	0.507	59.76	0.569	59.29	0.546
	ProtoNet	55.36	0.523	58.78	0.574	68.63	0.678	67.30	0.666	78.40	0.779	81.10	0.810
	Baseline	<b>73.56</b>	<b>0.727</b>	<b>84.57</b>	<b>0.845</b>	86.05	0.860	86.22	0.852	96.84	0.968	97.28	0.973
	Baseline++	72.33	0.716	82.84	0.824	<b>87.21</b>	<b>0.871</b>	<b>86.73</b>	<b>0.859</b>	<b>97.18</b>	<b>0.972</b>	<b>97.61</b>	<b>0.976</b>
UNI	KNN	63.57	0.618	72.39	0.718	73.70	0.719	87.54	0.866	96.99	0.970	97.16	0.971
	MatchingNet	42.64	0.410	43.80	0.350	46.83	0.383	61.17	0.578	51.86	0.430	50.90	0.440
	ProtoNet	54.03	0.513	62.45	0.615	62.32	0.619	69.70	0.694	79.52	0.793	80.11	0.798
	Baseline	<b>64.43</b>	<b>0.629</b>	80.16	0.800	82.85	0.828	88.18	0.872	98.08	0.981	98.39	0.984
	Baseline++	63.74	0.619	<b>80.35</b>	<b>0.802</b>	<b>82.96</b>	<b>0.829</b>	<b>88.36</b>	<b>0.874</b>	<b>98.37</b>	<b>0.984</b>	<b>98.76</b>	<b>0.988</b>

#### 4.2.6. Computational Complexity for Flexibility Assessment Scenario

Moreover, we evaluated the computational complexity of the four pathology-specific foundation models using various FSL methods. For each combination, we measured the number of parameters, the average execution time, and peak memory usage during training and testing per episode for 1-shot, 5-shot, and 10-shot settings (Table 10). Among the FSL methods, KNN, ProtoNet, and MatchingNet did not require any additional parameters, whereas **Baseline** and **Baseline++** resulted in a slight increase in the number of parameters, accounting for less than 0.1% of the original model size. Memory requirements varied depending on both the foundation models and FSL methods. Larger foundation models consumed more memory, and the FSL methods further contribute to the overall memory usage. For instance, ProtoNet and MatchingNet adjust all layers in the foundation models, leading to the highest memory consumption. In contrast, KNN, **Baseline**, and **Baseline++** do not consume memory during training, as KNN, **Baseline**, and **Baseline++** do not involve any training. During testing, since all parameters remained fixed, ProtoNet and MatchingNet demonstrate substantially reduced memory usage compared to the training phase for all four foundation models. Specifically, there was a 2.5 to 3.5 fold decrease for CTransPath, a 3.0 to 4.0 fold decrease for Lunit, a 4.3 to 5.8 fold decrease for Phikon, and a 7.8 to 10.8 fold decrease for UNI across different shot counts. Meanwhile, **Baseline** and **Baseline++** exhibited slightly increased memory usage compared to ProtoNet, MatchingNet, and KNN due to the modifications of the final linear layer. However, the execution time for the foundation models using **Baseline** and **Baseline++** was greatly slower compared to other FSL methods due to their design. With ProtoNet, MatchingNet, and KNN, the foundation models required less than 2.90 seconds to process one episode, whereas they took more than 202.00 seconds when using **Baseline** and **Baseline++**.

Table 9: Few-shot learning results on out-domain adaptation tasks.

		Prostate cancer grading						Gastric cancer sub-typing						Breast cancer detection														
Model	Method	1-shot			5-shot			10-shot			1-shot			5-shot			10-shot			1-shot			5-shot			10-shot		
		Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1		
CTransPath	KNN	44.67	0.421	65.40	0.637	68.55	0.662	62.81	0.617	72.22	0.720	75.45	0.750	55.26	0.545	62.42	0.624	71.39	0.697									
	MatchingNet	23.89	0.121	33.85	0.274	34.92	0.266	26.20	0.149	36.18	0.307	42.89	0.358	25.59	0.128	34.10	0.259	37.45	0.289									
	ProtoNet	27.66	0.254	32.60	0.298	32.09	0.291	32.17	0.307	46.97	0.452	51.43	0.497	35.09	0.328	38.19	0.368	46.27	0.436									
	Baseline	<b>46.73</b>	<b>0.439</b>	<b>74.44</b>	<b>0.737</b>	72.53	0.720	<b>66.68</b>	<b>0.654</b>	<b>78.93</b>	<b>0.789</b>	<b>84.82</b>	<b>0.848</b>	<b>57.40</b>	<b>0.569</b>	<b>72.78</b>	<b>0.723</b>	<b>79.15</b>	<b>0.788</b>									
	Baseline++	44.39	0.417	70.91	0.699	<b>73.46</b>	<b>0.728</b>	61.73	0.604	76.49	0.764	83.49	0.836	54.78	0.541	72.29	0.718	77.48	0.769									
Lunit	KNN	45.71	0.429	63.45	0.614	65.65	0.636	67.00	0.667	77.82	0.773	79.73	0.798	59.65	0.585	62.56	0.617	71.69	0.701									
	MatchingNet	22.54	0.219	25.97	0.200	26.85	0.190	35.35	0.340	38.08	0.317	40.75	0.329	38.77	0.370	44.76	0.406	40.75	0.335									
	ProtoNet	29.03	0.262	33.20	0.319	37.26	0.348	39.12	0.352	48.44	0.470	56.46	0.548	36.25	0.346	42.42	0.405	43.16	0.416									
	Baseline	<b>48.18</b>	<b>0.461</b>	72.03	0.712	70.18	0.693	68.50	0.681	79.69	0.798	86.03	0.860	<b>59.87</b>	<b>0.587</b>	<b>72.65</b>	<b>0.714</b>	<b>78.23</b>	<b>0.779</b>									
	Baseline++	45.95	0.435	<b>73.20</b>	<b>0.725</b>	<b>72.14</b>	<b>0.710</b>	<b>68.52</b>	<b>0.680</b>	<b>81.27</b>	<b>0.811</b>	<b>86.51</b>	<b>0.865</b>	58.90	0.578	72.28	0.711	77.37	0.770									
Phikon	KNN	45.81	<b>0.435</b>	64.75	0.616	67.88	0.656	66.27	0.657	71.95	0.715	75.85	0.757	<b>60.42</b>	<b>0.598</b>	65.05	0.644	72.27	0.717									
	MatchingNet	24.00	0.223	26.01	0.201	26.76	0.168	41.35	0.395	48.09	0.451	38.78	0.303	40.95	0.399	43.65	0.382	34.79	0.293									
	ProtoNet	29.53	0.268	33.65	0.325	35.12	0.337	38.46	0.367	48.83	0.477	58.81	0.581	41.92	0.388	42.48	0.402	51.66	0.487									
	Baseline	45.48	0.419	73.02	0.725	73.72	0.730	<b>67.71</b>	<b>0.675</b>	79.20	0.792	84.52	0.844	59.31	0.588	73.37	0.730	<b>80.28</b>	<b>0.802</b>									
	Baseline++	<b>45.94</b>	0.427	<b>75.28</b>	<b>0.746</b>	<b>75.30</b>	<b>0.745</b>	66.39	0.659	<b>80.29</b>	<b>0.803</b>	<b>85.41</b>	<b>0.854</b>	59.08	0.585	<b>74.44</b>	<b>0.739</b>	78.09	0.780									
UNI	KNN	47.02	0.427	63.32	0.610	66.20	0.631	<b>66.48</b>	<b>0.662</b>	76.29	0.763	81.60	0.815	<b>63.16</b>	<b>0.622</b>	71.17	0.707	79.41	0.787									
	MatchingNet	24.24	0.224	29.30	0.229	25.84	0.193	37.99	0.364	42.22	0.382	41.19	0.334	43.80	0.402	42.05	0.348	36.08	0.285									
	ProtoNet	32.19	0.295	29.90	0.274	37.16	0.356	35.26	0.339	50.63	0.485	50.77	0.491	34.52	0.319	37.53	0.357	43.82	0.420									
	Baseline	<b>47.96</b>	<b>0.433</b>	72.35	0.715	71.24	0.699	65.09	0.648	<b>82.32</b>	<b>0.824</b>	<b>89.34</b>	<b>0.894</b>	62.89	0.619	76.87	0.760	82.08	0.820									
	Baseline++	47.21	0.430	<b>73.44</b>	<b>0.726</b>	<b>74.10</b>	<b>0.728</b>	66.19	0.659	82.03	0.821	89.33	0.894	62.92	0.619	<b>78.22</b>	<b>0.774</b>	<b>82.94</b>	<b>0.830</b>									

Table 10: Computational complexity of foundation models and few-shot learning methods with varying shot counts.

Model	Method	# Params (M)	1-shot				5-shot				10-shot			
			Memory (MB)		Time (sec)		Memory (MB)		Time (sec)		Memory (MB)		Time (sec)	
			Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
CTransPath	KNN	27.523	-	3.675	-	1.83	-	3.817	-	2.26	-	3.773	-	2.86
	ProtoNet	27.523	9.319	3.675	2.27	1.82	11,049	3.817	2.28	2.26	13,123	3.773	2.85	2.27
	MatchingNet	27.523	9.319	3.675	2.66	2.35	11,049	3.817	2.70	2.37	13,123	3.773	2.68	2.80
	Baseline	27.526	-	3.683	-	204.67	-	3.781	-	205.14	-	3.825	-	205.12
	Baseline++	27.526	-	3.683	-	218.47	-	3.781	-	222.39	-	3.825	-	224.26
Lunit	KNN	21.667	-	2.183	-	1.82	-	2.243	-	1.82	-	2.373	-	1.83
	ProtoNet	21.667	6,641	2.183	2.28	1.83	7,777	2.243	2.71	1.86	9,293	2.373	2.69	2.70
	MatchingNet	21.667	6,641	2.183	1.83	1.87	7,759	2.243	2.27	1.92	9,297	2.373	2.66	2.37
	Baseline	21.669	-	2,191	-	203.42	-	2,251	-	203.35	-	2,381	-	205.14
	Baseline++	21.669	-	2,191	-	218.69	-	2,251	-	219.45	-	2,381	-	221.05
Phikon	KNN	85.802	-	2.871	-	1.83	-	2,951	-	2.26	-	3.103	-	2.27
	ProtoNet	85.802	12,263	2.871	2.28	1.86	14,493	2.951	2.30	2.30	17,893	3.103	2.71	2.25
	MatchingNet	85.802	12,263	2.871	1.86	2.38	14,493	2.951	2.30	2.40	17,893	3.103	2.84	2.83
	Baseline	85.805	-	2,879	-	203.53	-	2,959	-	208.41	-	3,111	-	209.38
	Baseline++	85.805	-	2,879	-	203.20	-	2,959	-	203.32	-	3,111	-	206.58
UNI	KNN	303.355	-	3.677	-	1.84	-	3,683	-	1.85	-	3.897	-	1.85
	ProtoNet	303.355	28,680	3.677	3.63	1.84	34,161	3.683	4.11	1.87	42,000	3.897	4.16	2.31
	MatchingNet	303.355	28,680	3.677	2.61	1.94	34,161	3.683	2.72	2.39	42,000	3.897	4.13	2.39
	Baseline	303.359	-	3,685	-	203.35	-	3,691	-	203.53	-	3,905	-	205.81
	Baseline++	303.359	-	3,685	-	202.77	-	3,691	-	204.09	-	3,905	-	203.61

## 5. Discussion

This study provides a comprehensive evaluation of pathology-specific foundation models, assessing their adaptability across various datasets and tasks through consistency and flexibility assessment scenarios. The experimental results offer key insights into how these foundation models can be effectively fine-tuned and applied to various pathology tasks under diverse conditions.

Our analyses revealed that UNI exhibited the most stable performance among the pathology-specific four models in the consistency assessment scenario. The success of UNI is primarily attributable to its larger model size and extensive training data. UNI employed the ViT-L architecture as its backbone and was pretrained using DINoV2 on the MASS-100k dataset. With 303.355 million parameters, UNI has the largest number of parameters among the four foundation models and was trained on the most extensive dataset. This combination of vast data and numerous parameters played a critical role in obtaining superior generalization performance compared to other foundation models. However, the large model size introduces a trade-off between performance and complexity. Due to the extensive parameter count, UNI consumes more computational resources during training compared to other foundation models. Given the continuous increase in dataset sizes across numerous tasks in pathology, the high complexity of UNI can be a practical concern. This issue can be somewhat mitigated through the use of PEFT. PEFT showed superior performance in the consistency assessment scenario across various datasets and tasks compared to other adaptation strategies. Strength of PEFT lies in its ability to achieve excellent performance while minimizing computational resources, as opposed to traditional fine-tuning methods such as linear probing or full fine-tuning. This makes PEFT a highly efficient and effective approach for both development and deployment in real-world clinical environments.

In the flexibility assessment scenario, FSL methods such as **Baseline** and **Baseline++** proved effective in adapting pathology-specific foundation models to new tasks with limited data. These methods preserved capabilities of pre-trained models while efficiently adapting to new, unseen data, requiring only minimal modifications to the original models. On the other hand, approaches that involve fine-tuning the entire model, like MatchingNet and ProtoNet, performed worse than KNN. As training data is extremely limited (i.e., 1-shot out-domain scenarios), KNN sometimes outperformed **Baseline** and **Baseline++**, indicating that the prior knowledge of the foundation models is sufficient to handle new tasks. However, as the number of shots increases, the adoption of **Baseline** and **Baseline++** provided superior performance. This suggests that while the foundation models can handle new tasks with minimal data, they still benefit from additional data and specialized FSL methods to further enhance their performance.

Comparing the computational complexity of the four pathology-specific foundation models, the test time per image (equivalent to the test time with a batch size of 1) was similar across foundation models, whereas the test time per batch varied. This difference is attributable to the computational efficiency in handling larger quantities of data in a batch. With smaller batch sizes, all intermediate values generated during model operations can be stored in on-chip memory, reducing the need for off-chip memory access and thus minimizing any significant increase in processing time across foundation models. However, as larger batch sizes grow and models become larger, the test time surges significantly, likely due to factors such as memory bandwidth and hierarchy, the size of intermediate values, and data access patterns. For example, the test time (and training time) per batch for UNI was more than nine times higher than that for Lunit.

The larger model size of UNI needs processing a greater number of parameters and intermediate values, requiring more frequent off-chip memory accesses, which leads to increased processing time. Conversely, Lunit demonstrated a relatively smaller increase in execution time per batch, owing to its smaller number of parameters and lower memory usage. Thus, more complex models experience a sharp rise in processing time with larger batch sizes, whereas the difference in execution time between models remains minimal with smaller batch sizes. These observations are consistent with the previous findings described in [61].

Our study emphasizes the importance of selecting appropriate adaptation strategies for pathology-specific foundation models. Fine-tuning while preserving existing knowledge is crucial for effectively successfully applying the foundation models to a variety of tasks. Though some studies highlighted the zero-shot capabilities of foundation models, particularly visual-language foundation models [62, 10], the results in the flexibility assessment scenarios suggest that utilizing additional small data is still beneficial. FSL methods, however, have not yet been fully studied and developed for pathology-specific foundation models, presenting an area for further exploration. Moreover, the size of foundation models tends to increase, so do the execution time and memory requirements for both training and testing, which could pose a significant barrier to their practical use in clinical settings, in regard to the enormous number of tasks in pathology.

There are limitations in our study. First, our study is limited to patch-level pathology image classification, which does not encompass various aspects of pathology image analysis. It is necessary to expand our study to other tasks such as whole slide-level classification [63], cell segmentation [64], and etc. Second, five fine-tuning and five FSL methods are employed in this study. While PEFT and **Baseline/Baseline++** consistently showed advantages in this benchmark, performance varied depending on the specific foundation model and dataset/task. For instance, although PEFT generally yielded superior results, partial fine-tuning still showed competitive performance in certain tasks, such as prostate cancer grading. Optimizing a foundation model for a particular dataset/task requires a further investigation. This issue is beyond the scope of this study and will be explored in future research. Third, in the flexibility assessment scenario, colorectal tissue datasets are used for  $X_{Meta-train}$  and  $X_{Meta-val}$ , while  $X_{Meta-test}$  includes colon, lung, prostate, gastric, and breast tissues. The relationship between colorectal tissues with other tissue types may vary, which could influence the performance in adaptation tasks. We leave further exploration of this issue for future study. Last, while the consistency assessment scenario adopts multiple dataset per adaptation task, the flexibility assessment scenario involves a single dataset for each specific adaptation task. To further validate our findings on the flexibility assessment of the foundation models, an extended validation study, including multiple external datasets, needs to be followed.

## 6. Conclusion

In this study, we benchmarked four pathology-specific foundation models, focusing on adaptation strategies and performance across various scenarios. Fine-tuning foundation models, particularly using PEFT, proved effective in addressing data variabilities due to differences in data source and acquisition settings within the same downstream tasks. PEFT consistently outperformed other fine-tuning methods by maintaining superior performance across diverse datasets/tasks while demonstrating comparable computational complexity. Moreover, the strength and usefulness of foundation models were validated

in data-limited environments. Foundation models with KNN demonstrated the ability to adapt to unseen tasks, leveraging the prior knowledge of foundation models in pathology. The integration of specialized FSL methods, such as **Baseline** and **Baseline++**, could further enhance the generalization ability of foundation models, enabling more robust performance across various adaptation tasks. Future study will focus on developing more advanced fine-tuning and FSL methods tailored to pathology-specific foundation models and optimizing them to further improve both performance and computational complexity.

## Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) (No. 2021R1A2C2014557 and No. RS-2024-00397293) and by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program (No. P0022543).

## References

- [1] Shaoting Zhang and Dimitris Metaxas. On the challenges and perspectives of foundation models for medical image analysis. *Medical Image Analysis*, page 102996, 2023.
- [2] 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.
- [3] Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical image analysis*, 58:101563, 2019.
- [4] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical image analysis*, 81:102559, 2022.
- [5] Mingu Kang, Heon Song, Seonwook Park, Donggeun Yoo, and Sérgio Pereira. Benchmarking self-supervised learning on diverse pathology datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3344–3354, 2023.
- [6] Alexandre Filiot, Ridouane Ghermi, Antoine Olivier, Paul Jacob, Lucas Fidon, Alice Mac Kain, Charlie Saillard, and Jean-Baptiste Schiratti. Scaling self-supervised learning for histopathology with masked image modeling. *medRxiv*, pages 2023–07, 2023.
- [7] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3):850–862, 2024.

- [8] Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Siqi Liu, Philippe Mathieu, Alexander van Eck, Donghun Lee, Julian Viret, et al. Virchow: a million-slide digital pathology foundation model. *arXiv preprint arXiv:2309.07778*, 2023.
- [9] Jonas Dippel, Barbara Feulner, Tobias Winterhoff, Simon Schallenberg, Gabriel Dernbach, Andreas Kunft, Stephan Tietz, Philipp Jurmeister, David Horst, Lukas Ruff, et al. Rudolfv: a foundation model by pathologists for pathologists. *arXiv preprint arXiv:2401.04079*, 2024.
- [10] 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.
- [11] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. Review the cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary Oncology/Współczesna Onkologia*, 2015(1):68–77, 2015.
- [12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [14] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.
- [15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.
- [17] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.
- [18] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11205–11214, 2021.

- [19] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.
- [20] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *Advances in neural information processing systems*, 33:5679–5690, 2020.
- [21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [22] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*, 2020.
- [23] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5652–5660, 2021.
- [24] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [25] Yang Luo, Zhineng Chen, Shengtian Zhou, and Xieping Gao. Self-distillation augmented masked autoencoders for histopathological image classification. *arXiv preprint arXiv:2203.16983*, 2022.
- [26] Trinh Thi Le Vuong, Quoc Dang Vu, Mostafa Jahanifar, Simon Graham, Jin Tae Kwak, and Nasir Rajpoot. Impash: A novel domain-shift resistant representation for colorectal cancer tissue classification. In *European Conference on Computer Vision*, pages 543–555. Springer, 2022.
- [27] Navid Alemi Koohbanani, Balagopal Unnikrishnan, Syed Ali Khurram, Pavitra Krishnaswamy, and Nasir Rajpoot. Self-path: Self-supervision for classification of pathology images with limited annotations. *IEEE Transactions on Medical Imaging*, 40(10):2845–2856, 2021.
- [28] Chetan L Srinidhi, Seung Wook Kim, Fu-Der Chen, and Anne L Martel. Self-supervised driven consistency training for annotation efficient histopathology image analysis. *Medical image analysis*, 75:102256, 2022.
- [29] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16144–16155, 2022.
- [30] Yoo Jung Kim, Hyungjoon Jang, Kyoungbun Lee, Seongkeun Park, Sung-Gyu Min, Choyeon Hong, Jeong Hwan Park, Kanggeun Lee, Jisoo Kim, Wonjae Hong, et al. Paip 2019: Liver cancer segmentation challenge. *Medical image analysis*, 67:101854, 2021.

- [31] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.
- [32] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [33] Zhengliang Liu, Yiwei Li, Peng Shu, Aoxiao Zhong, Longtao Yang, Chao Ju, Zihao Wu, Chong Ma, Jie Luo, Cheng Chen, et al. Radiology-llama2: Best-in-class large language model for radiology. *arXiv preprint arXiv:2309.06419*, 2023.
- [34] Botond Fazekas, José Morano, Dmitrii Lachinov, Guilherme Aresta, and Hrvoje Bogunović. Adapting segment anything model (sam) for retinal oct. In *International Workshop on Ophthalmic Medical Image Analysis*, pages 92–101. Springer, 2023.
- [35] Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1114, 2024.
- [36] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.
- [37] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.
- [38] 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.
- [39] 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.
- [40] Jessica Deuscher, Daniel Firnbach, 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.
- [41] Fereshteh Shakeri, Malik Boudiaf, Sina Mohammadi, Ivaxi Sheth, Mohammad Havaei, Ismail Ben Ayed, and Samira Ebrahimi Kahou. Fhist: A benchmark for few-shot classification of histological images. *arXiv preprint arXiv:2206.00092*, 2022.
- [42] Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019.

- [43] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6(1):1–11, 2016.
- [44] Sajid Javed, Arif Mahmood, Muhammad Moazam Fraz, Navid Alemi Koohbanani, Ksenija Benes, Yee-Wah Tsang, Katherine Hewitt, David Epstein, David Snead, and Nasir Rajpoot. Cellular community detection for tissue phenotyping in colorectal cancer histology images. *Medical image analysis*, 63:101696, 2020.
- [45] Trinh Thi Le Vuong, Kyungeun Kim, Boram Song, and Jin Tae Kwak. Joint categorical and ordinal learning for cancer grading in pathology images. *Medical image analysis*, 73:102206, 2021.
- [46] Chuang Zhu, Wenkai Chen, Ting Peng, Ying Wang, and Mulan Jin. Hard sample aware noise robust learning for histopathology image classification. *IEEE transactions on medical imaging*, 41(4):881–894, 2021.
- [47] Qian Da, Xiaodi Huang, Zhongyu Li, Yanfei Zuo, Chenbin Zhang, Jingxin Liu, Wen Chen, Jiahui Li, Dou Xu, Zhiqiang Hu, et al. Digestpath: A benchmark dataset with challenge review for the pathological detection and segmentation of digestive-system. *Medical Image Analysis*, 80:102485, 2022.
- [48] Wouter Bulten, Kimmo Kartasalo, Po-Hsuan Cameron Chen, Peter Ström, Hans Pinckaers, Kunal Nagpal, Yuannan Cai, David F Steiner, Hester Van Boven, Robert Vink, et al. Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge. *Nature medicine*, 28(1):154–163, 2022.
- [49] Xinmi Huo, Kok Haur Ong, Kah Weng Lau, Laurent Gole, David M Young, Char Loo Tan, Xiaohui Zhu, Chongchong Zhang, Yonghui Zhang, Longjie Li, et al. A comprehensive ai model development framework for consistent gleason grading. *Communications Medicine*, 4(1):84, 2024.
- [50] Guy Nir, Soheil Hor, Davood Karimi, Ladan Fazli, Brian F Skinnider, Peyman Tavassoli, Dmitry Turbin, Carlos F Villamil, Gang Wang, R Storey Wilson, et al. Automatic grading of prostate cancer in digitized histopathology images: Learning from multiple experts. *Medical image analysis*, 50:167–180, 2018.
- [51] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part II 11*, pages 210–218. Springer, 2018.
- [52] 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.

- [53] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, et al. Bracs: A dataset for breast carcinoma subtyping in h&e histology images. *Database*, 2022:baac093, 2022.
- [54] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, et al. Bach: Grand challenge on breast cancer histology images. *Medical image analysis*, 56:122–139, 2019.
- [55] Jaeung Lee, Chiwon Han, Kyungeun Kim, Gi-Ho Park, and Jin Tae Kwak. Camelnet: centroid-aware metric learning for efficient multi-class cancer classification in pathology images. *Computer Methods and Programs in Biomedicine*, 241:107749, 2023.
- [56] Andrew A Borkowski, Marilyn M Bui, L Brannon Thomas, Catherine P Wilson, Lauren A DeLand, and Stephen M Mastorides. Lung and colon cancer histopathological image dataset (lc25000). *arXiv preprint arXiv:1912.12142*, 2019.
- [57] Christian Abbet, Linda Studer, Andreas Fischer, Heather Dawson, Inti Zlobec, Behzad Bozorgtabar, and Jean-Philippe Thiran. Self-rule to adapt: Learning generalized features from sparsely-labeled data using unsupervised domain adaptation for colorectal cancer tissue phenotyping. *Proceedings of the Medical Imaging with Deep Learning (MIDL 2021), 7-9 July 2021, Lübeck, Germany*, 2021.
- [58] Trinh Thi Le Vuong and Jin Tae Kwak. Moma: Momentum contrastive learning with multi-head attention-based knowledge distillation for histopathology image analysis. *arXiv preprint arXiv:2308.16561*, 2023.
- [59] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.
- [60] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knnc model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer, 2003.
- [61] Jongsoo Park, Maxim Naumov, Protonu Basu, Summer Deng, Aravind Kalaiah, Daya Khudia, James Law, Parth Malani, Andrey Malevich, Satish Nadathur, et al. Deep learning inference in facebook data centers: Characterization, performance optimizations and hardware implications. *arXiv preprint arXiv:1811.09886*, 2018.
- [62] 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.
- [63] Kyungmo Kim, Kyoungbun Lee, Sungduk Cho, Dong Un Kang, Seongkeun Park, Yunsook Kang, Hyunjeong Kim, Gheeyoung Choe, Kyung Chul Moon, Kyu Sang Lee, et al. Paip 2020: Microsatellite instability prediction in colorectal cancer. *Medical Image Analysis*, 89:102886, 2023.

- [64] Tan NN Doan, Boram Song, Trinh TL Vuong, Kyungeun Kim, and Jin T Kwak. Sonnet: A self-guided ordinal regression neural network for segmentation and classification of nuclei in large-scale multi-tissue histology images. *IEEE Journal of Biomedical and Health Informatics*, 26(7):3218–3228, 2022.