1 Datasets

1.1 Pathology image classification tasks

The datasets used to evaluate our framework are collected from two distinct pathology projects [35, 39]. Both projects make their data available as patches, which are extracted from H&E stained Whole Slide Images (WSI) digitized from Formalin-Fixed Paraffin-Embedded (FFPE) slides. Each dataset aims to detect different cancerous tissues. The stain used in both datasets is hematoxylin and eosin (H&E), a commonly-used stain.

PatchCamelyon (PCam) [35] classification. PCam is a patch dataset extracted from Camelyon16 challenge [2]. The original 400 WSIs from Camelyon16 were digitized breast tissue with potential metastasized cancerous tissue on the lymph nodes. The original WSIs were scanned at 40× resolution but later downsampled to 10×. The WSIs were collected from two different centers. PCam extracted 327,680 patches at 96×96 resolution and labeled them positive or negative. Positive labeled patches present tumor tissue in the central 32×32 patch region. There are an equal amount of positive and negative labeled samples.

MHIST [39] classification. MHIST contains 3,152 patches from colorectal regions at 224×224 pixel resolution. These patches were extracted from 328 WSIs scanned at 40× resolution. Each patch may be labeled Hyperplastic Polyp (HP) or Sessile Serrated Adenoma (SSA). HP is the majority class with 68.59% of labels. Labeling colorectal polyps between HP and SSA is challenging due to high inter-pathologist disagreement. Seven pathologists contributed to the ground truth to ensure reliable labels.

1.2 GM/WM segmentation task

Besides the two pathology classification tasks above, we follow [19] and evaluate our proposed approach on a grey/white matter segmentation task in pathological slides. For this dataset, we have 30 gigapixel slides with an average resolution of $50,000 \times 60,000$ by an Aperio AT2 scanner at 20X magnification. We follow the previous work [19] to split 20 slides into the training/validation set while 10 slides into the hold-out test set. We tile these gigapixel slides into patches at 256×256 and convert it into a weakly-supervised task [23].

1.3 Multi-task objective in pathology image analysis

The digital pathology domain mandates the execution of multiple tasks for comprehensive analysis, encompassing segmentation, classification, and/or detection tasks, as shown in Fig 5. Therefore, proposing an efficient method to swiftly adapt a large-scale model to multiple downstream tasks would be exceptionally advantageous, enabling the attainment of the final analysis objective within a reasonable timeframe. It is impractical if the fine-tuning requires heavy computational resources when the learning tasks are separated from the analysis objective.

As shown in Fig 5: multi-tasks need to be performed for the final analysis of the digital pathology domain: this may involve segmentation, classification, and/or detection tasks. For example, a recent work [19] quantifies and visualizes each type of pathology in grey and white matter regions in gigapixel images, respectively. To achieve such an objective, we need to separate it into two learning tasks: pathology detection and grey/white matter (GM/WM) segmentation. They collected and annotated two datasets, then trained two models from scratch to achieve it. Therefore, it will be extremely beneficial if we can propose an effective way to adapt a big model into multiple downstream tasks quickly so the final analysis objective can be reached in a reasonable time manner.

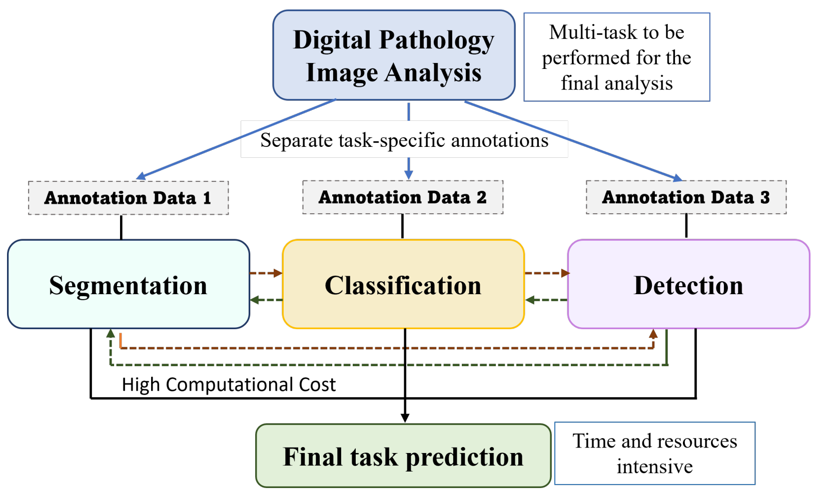


Fig. 5: Overview of the pathology image analysis: it may involve various vision tasks.

2 Path-CLIP: Main Algorithm

Language-vision training in medical domain. Language-vision pre-training typically requires near-infinite web images and captions from general domains, e.g., CLIP uses 400 million image-caption pairs [29]. Such datasets dwarf the scale of medical datasets. Another challenge is the annotation cost: medical images typically require domain knowledge from trained personnel [20]. This makes the training cost-expensive if we have multiple medical tasks. For example, Lai *et al.* studies the distribution of Amyloid-β plaques (a hallmark pathology with Alzheimer's disease) in grey and white matter, which involves two learning tasks: image segmentation and object detection. These two tasks require two different datasets with their required annotation to train two models [19].

Although expert annotation of medical images can be expensive, preparing the captions and prompts for images will exacerbate the situation if we use language-

Algorithm 1 Path-CLIP: Adapt CLIP for pathology tasks.

Input: N: size of the dataset, $F(\cdot)$: image encoder, $G(\cdot)$: text encoder, x: input image, t: input text, $w(\cdot)$: weak augmentation, $s(\cdot)$: strong augmentation.

Freeze $F(\cdot)$ and $G(\cdot)$
for epoch $= 1, 2, \ldots$ **do**
    for image $= x$ **do**
        Extract image features: $f = F(x)$
        Generate the Gaussian noise ϵ
        Perturbed the image features: $\hat{f} = f + \epsilon$
        Fine-tune $\hat{f}$ with TFT module: $L(\hat{f})$
        for $i = 1, 2, ..$ **do**
            Compute self-adaptive residual ratio from (7):
            $\alpha_i = \min(\max(\tau, T), 1 - \tau)$,
            where, T→Accuracy, and
            $\tau = \frac{1}{N} \sum_{i=1}^{N} cos(F(w(x_i)), F(s(x_i)))$
            calculate $f_i^* = (1 - \alpha_i) L(\hat{f}_i) + \alpha_i f_i$
        end for
    end for
    Perform text embedding: $g = G(t)$
    Define the contrastive loss l_i^{c} from (8)
    Optimize (9) for Doublet Multimodal Contrastive Loss (DMCL):
      $\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{CLIP}} + \mathcal{L}_{\mathrm{SWAPP}}$, where,
        $\mathcal{L}_{\mathrm{CLIP}} \rightarrow \frac{1}{2B} \sum_{i=1}^{B} \left(l_i^{v \rightarrow t} + l_i^{t \rightarrow v} \right)$,
        $\mathcal{L}_{\mathrm{SWAPP}} \rightarrow \frac{1}{2B} \sum_{i=1}^{2B} l_i^{\mathrm{c}}$, B is the mini-batch size of the image-text pair.
end for

vision training. For example, MedCLIP [38] used 570 thousands of image-text pairs to achieve 60% zero-shot accuracy. Datasets with such scale are challenging to curate in pathology image tasks, where each slide is at gigapixel level. The efforts to adapt CLIP for the pathology domain remain limited due to the large gap between the general and medical image domains. In this work, we aim to design an efficient adaption framework that can be scalable for multiple downstream tasks.

Path-CLIP: Tailored Adaptation for Pathology In digital pathology, scanning and slide preparation may introduce artifacts such as staining artifacts, tissue folds, or scanner-related distortions. These artifacts can affect feature extraction and analysis. Moreover, pathology images often contain sparse, intricate, and irregular features. Identifying specific cells, structures, or abnormalities within these images is challenging, especially when adapting to a different domain, specifically non-medical. RFR, a shallow network, exhibits the capability to extract spatial features and patterns from pathology images, aiding in the optimization of feature extraction and adaptation processes. In RFR, we design a self-adaptive module to automatically fuse/balance the original knowledge from CLIP and the new knowledge learned from the new pathological task with only a few additional trainable weights instead of optimizing the entire encoders in CLIP. To further alleviate the overfitting, we also incorporate Hidden Representation Perturbation (HRP) and Dual-view Vision Contrastive (DVC). Lastly, we build an end-to-end adaptation of CLIP with a proposed loss, Doublet Multimodal Contrastive Loss (DMCL), to pathology tasks, named Path-CLIP. The main algorithm is summarized in Algorithm 1

3 More Ablation Studies

Table 6: Quantitative results on the hold-out test set of PCam.

Algorithm	Data Usage	Accuracy	Recall	Precision	F1-score	AUC
CLIP [29]	Zero-shot	56.5	50.3	57.4	53.7	0.600
CLIP + RFR	0.1%	76.4	90.0	70.7	79.2	0.849
	0.5%	81.5	85.0	79.4	82.1	0.894
	1%	81.9	82.9	81.3	82.1	0.900
	5%	82.9	77.1	87.2	81.8	0.918
	10%	82.8	79.2	85.4	82.1	0.914
	50%	81.4	71.0	89.6	79.3	0.918

3.1 How does RFR improve CLIP?

In this subsection, we discuss the results on PCam [35] and MHIST [39] to show how RFR's fine-tuning improves CLIP. We select Accuracy, Recall, Precision, F1-score, and AUC to compare comprehensively. The results are summarized

Table 7: Quantitative results on the hold-out test set of MHIST.

Algorithm	Data Usage	Accuracy	Recall	Precision	F1-score	AUC
CLIP [29]	Zero-shot	36.9	100.0	36.9	53.9	0.501
CLIP + RFR	1%	63.9	7.5	57.5	13.3	0.643
	5%	66.8	42.8	56.6	48.7	0.732
	10%	70.5	79.7	57.1	66.6	0.784
	20%	70.7	86.1	56.8	68.4	0.788
	50%	74.8	75.6	63.3	68.9	0.838

in Table 6 and Table 7. We show that RFR can bring over 25% improvement in accuracy and 28.4% in F1-score by only using 0.5% of data in PCam [35] compared to the original CLIP. We get significant improvement in accuracy and AUC on MHIST [39], while the performance in F1-score and recall seems limited. We diagnose that MHIST as a more challenging task with an inter-rate agreement issue, which may confuse the model during fine-tuning.

Table 8: Vision vs prompt fine-tuning on PCam.

Data Usage	Algorithm	Accuracy	Training Time
Zero-shot	CLIP [29]	56.5	-
0.1%	CoOp [59]	58.5	**7 min 6 sec**
	Path-CLIP	**76.4**	10 min 29 sec
1%	CoOp [59]	61.5	53 min 21 sec
	Path-CLIP	**81.9**	**11 min 56 sec**
10%	CoOp [59]	59.9	2 h 23 min 45 sec
	Path-CLIP	**83.0**	**27 min 18 sec**

3.2 Learning rate of CLIP's directly fine-tuning

Fig. 6(a) summarizes the results when we change the learning rate during direct fine-tuning: CLIP's knowledge can be forgotten when it is large but the learning ability can be trivial if it is too small.

Moreover, we summarize our ablations on epochs and frozen layers in Table 9: the learning rate has been optimized for each setting. CLIP is pre-trained on 400M data, traditional fine-tuning methods can easily hurt its pre-training's generalizability.

3.3 Forgetting metric.

We propose to use ImageNet zero-shot accuracy as the forgetting metric. As shown in Table 10, our method can preserve more pre-trained knowledge when adapting CLIP on PCam and MHIST.

Table 9: Ablation on fine-tuning CLIP (ResNet-50) on PCam.

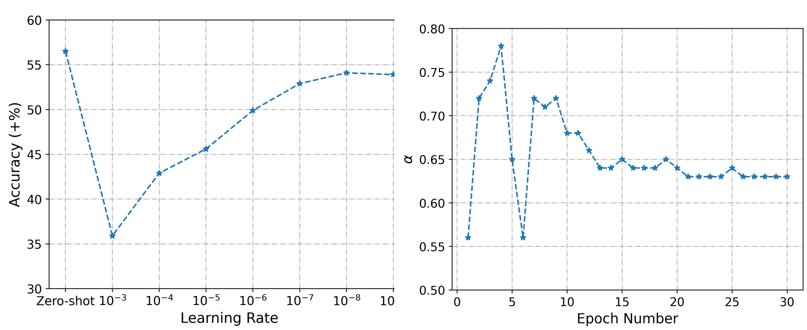
#	Epoch	Layer	Accuracy
1	zero-shot	-	56.5%
2	5 epochs	Whole Model	45.3%
3	10 epochs	Whole Model	40.3%
4	50 epochs	Whole Model	38.9%
5	5 epochs	Last Layer	54.2%
6	5 epochs	Last 5-Layer	52.2%

Table 10: ImageNet zero-shot accuracy as the forgetting metric. We use PCam and MHIST as our target task.

Methods	PCam	MHIST
Zero-shot CLIP	62.53%	62.53%
Fine-tune	42.56%	24.32%
CLIP-Adapter	44.32%	28.21%
Path-CLIP (Ours)	53.43%	53.58%

3.4 Ablation on self-adaptive ratio α

The self-adaptive residual ratio via epochs is visualized in Fig. 6(b): α tends to be small at the very beginning to learn more new knowledge, which may introduce overfitting, then α tends to increase and eventually stays stable.



(a) Learning Rate (b) Self-adaptive Residual Ratio

Fig. 6: Ablation studies on learning rate for traditional fine-tuning, and self-adaptive residual ratio α on PCam.

3.5 Vision vs prompt fine-tuning

CoOp [59] is the recent state-of-the-art fine-tuning method for CLIP. Hence, we mainly compare our proposed RFR with its performance and computational complexity. We summarize the results in Table 8. When we range the data usage from 0.1% to 1%, we reveal Path-CLIP can obtain 5.5% improvement in accuracy while only introducing 1 minute of additional training time. However, CoOp [59] has the overfitting issue and obtains a lower score while using 53 minutes, which is almost five times compared to the time used by RFR. The situation remains similar under the 10% of data usage. We conclude our proposed RFR can get over 25% improvement on CLIP by only using 10 minutes for the fine-tuning on a single GPU, which is promising for digital pathology research. Table 8 demonstrates results revealing vision fine-tuning for a medical domain may be more effective than prompt fine-tuning.

3.6 Computational efficiency

We summarize the results in Table 8. When we range the data usage from 0.1% to 1%, we reveal Path-CLIP can obtain 5.5% improvement in accuracy while only introducing 1 minute of additional training time. However, CoOp [59] has the overfitting issue and obtains a lower score while using 53 minutes, almost five times compared to the time used by RFR. The situation remains similar under the 10% of data usage. We conclude our proposed RFR can get over 25% improvement on CLIP by only using 10 minutes for the fine-tuning on a single GPU, which is promising for digital pathology research. Table 8 demonstrates that vision fine-tuning for a medical domain may be more effective than prompt fine-tuning.

3.7 Generalization to PLIP

PLIP [15] is pre-trained on a large-scale pathology dataset and achieves strong zero-shot performance. With our method, we can further boost the performance and outperform other tuning methods. The results are summarized in Table 11.

Table 11: Generalization on PLIP pre-trained weights when using different amounts of data for PCam adaptation.

Pre-trained	Algorithm	Zero-shot	0.1%	0.5%	1%	5%
PLIP	-	72.4	-	-	-	-
PLIP	CoOp	-	73.2	73.4	73.5	73.9
PLIP	CLIP-Adapter	-	75.4	78.3	80.2	80.3
PLIP	TIP-Adapter	-	76.1	77.9	79.1	79.2
PLIP	Path-CLIP (Our)	-	**79.4**	**83.5**	**86.7**	**86.5**

3.8 Generalization to other medical modality

We also evaluated our method on two X-ray datasets (CheXpert [16] and COVID [30]). The results are summarized in Table 12: we change the prompt as “this is a X-ray of []”. To show the generalizability of our approach, we also tried FLIP and PLIP: when only using 0.5% of PCam, FLIP can be improved from 59.5% to **76.5%** via our proposed Residual Feature Refinement (RFR). The results for PLIP can be found in Table 9 in the Appendix. We will move them to the main paper to show its generalizability.

Table 12: Generalization on X-ray datasets (accuracy).

Algorithm	Data Usage	CheXpert	COVID
CLIP	Zero-shot	20.16	50.69
CoOp	0.1%	23.59	54.62
CLIP-Adapter	0.1%	32.56	70.32
Path-CLIP (Our)	0.1%	**42.65**	**76.22**