Learning Heterogeneous Tissues with Mixture of Experts for Gigapixel Whole Slide Images

Supplementary Material

In this supplementary material, we first provide a notation table for the method section in Tab. 4. Then, we offer Details on Integrating PAMoE with several classical methods in Sec. 8. Following that, we include Implementation Details of using CONCH as a Classifier in Sec. 9. In Sec. 10, we present Datasets and Implementation Details. In Sec. 11, we provide Further Ablation Studies. Finally, in Sec. 12, we showcase Experiments on More Tasks to further demonstrate its effectiveness.

7. Notations

Table 4. Notation Table.

Symbol	Description
n	The total number of patches from a input set from a WSI.
N	The total number of patches generated from all WSIs in certain cancer dataset.
$\widetilde{N}$	The number of patches randomly sampled from the set of N patches.
H	The feature set of all WSIs in certain cancer dataset.
$\overline{H}$	The feature set randomly sampled from the overall feature set.
$\overline{S}$	Assignment score between experts and instances.
S	Assignment probability (Softmax normalized assignment score) between experts and instances.
Ω	The set of selected categories.
$\mathcal{P}$	The set of pre-extracted prior-based prototypes.
m	The total number of experts.
c	The capacity factor of PAMoE.

8. Details on Integrating PAMoE

This section provides a detailed explanation of integrating PAMoE with classical methods in Sec. 4.3.

TransMIL [36]. TransMIL is a classical transformer-based method that uses Nystrom attention [40] to reduce the computational complexity of self-attention. Additionally, it propose the Pyramid Position Encoding Generator (PPEG) module, which enables TransMIL to be aware of spatial information. Since TransMIL relies solely on self-attention to explore the morphological interactions between patches without an FFN layer following it, we inserted the PAMoE module after the self-attention layer while keeping the other modules unchanged. Since TransMIL employs the PPEG module to model positional relationships among patches, the arrangement of patches is meaningful. Therefore, the all-zero features of discarded patches are all retained.

LongVit [39]. LongViT is a Vision Transformer (ViT) framework designed to handle massive token inputs. It uses Dilated Attention to reduce the computational time and memory consumption associated with the massive number of tokens during self-attention calculations. With LongViT, we can build the ViT model with more parameters capable of handling tens of thousands of tokens, even with limited computational resources. We selected LongViT as one of the baseline models to verify the performance of a ViT with a larger-scale parameter set in WSI analysis tasks compared to other classical models. When integrating PAMoE with LongViT, we adopted the most commonly used method of integrating MoE within the transformer architecture by replacing the FFN layer with an MoE layer. Following recent practice [8, 25], we replace the feed-forward component of every other Transformer layer with a PAMoE layer, and interleave regular Transformer layers and MoE layers.

PatchGCN [5]. PatchGCN is a classical MIL model used for analyzing WSI. It employs a GNN to model the spatial relationships between patches, enabling the model to be context-aware. Unlike transformer-based models that use self-attention for global message passing, the characteristics of the GNN enable feature interactions within PatchGCN to occur over a more localized range. In this way, we test the types of models suitable for PAMoE and formulate several hypotheses. In terms of implementation, PatchGCN first uses an MLP layer to map the features of patches to a lower dimension, and then employs a GNN to perform message passing among adjacent patches. We directly replace the MLP layer with PAMoE layer, allowing different experts to map the heterogeneous patches, while keeping the other components unchanged.

9. Implementation Detail of using CONCH as Classifier

This section provides a detailed explanation of using CONCH as a classifier in Sec. 4.2. CONCH [30] is a visual-

language foundation model. The model can be immediately used for downstream classification tasks due to the aligned visual-language pretraining, which eliminates the need for additional labeled examples for supervised learning or fine-tuning. As shown in Fig. 6, we use text prompts to map patches and prompts into the same embedding space, comparing the cosine similarity of the representations to obtain category labels. We constructed our set of predetermined text prompts based on the set of class or category names provided by CONCH. The set of categories and their names we use including: 0. lymphoid infiltrate 1. stroma 2. tumor 3. necrosis 4. others (adipose, background, penmarking, mucin, muscle, benign epithelium)

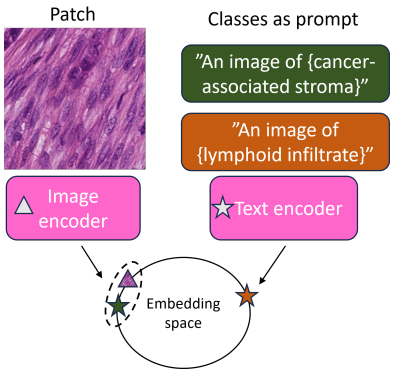

Figure 6. Schematic of zero-shot classification using contrastively aligned image and text encoders of CONCH.

10. Datasets and Implementation Details

This section provides a detailed explanation of the datasets and implementation details for experiments in Sec. 5.

10.1. Datasets

We conducted experiments based on survival task across different cancer types. All cancer types of WSIs are from The Cancer Genome Atlas (TCGA) repository.[1] We choose these cancer types for training and evaluation using the following criteria: 1) overall label available, 2) balanced distribution of uncensored-to-censored patients in survival task During dataset construction, we only preserve formalin-fixed paraffin-embedded hematoxylin and eosin (H&E) slides, considering the morphological alternation in frozen sections. For all approaches within the same task, we use the same 5-fold cross-validation splits. The specific split settings are provided in the accompanying code. We validated PAMoE on the survival prediction task across five cancer types, including: Breast Invasive Carcinoma (BRCA) (999 cases), Lower Grade Glioma (LGG) (773 cases), Lung Adenocarcinoma (LUAD) (492 cases), Colon Adenocarcinoma (COAD) (400 cases), Pancreatic Adenocarcinoma (PAAD) (192 cases). We use the concordance index (C-index) for the evaluation of all cancer types.

[1] https://portal.gdc.cancer.gov/

10.2. Implementation Details

Patch Extraction and Embedding. First, we use OTSU to extract foreground tissue regions. Then we extract a series of non-overlapping patches at $20\times$ magnification with size 256×256 which contain more than 50% foreground tissue. All patches are encoded by UNI [6] into 1024-dimensional vectors, and the encoder does not perform data augmentation during inference.

Network Hyper-Parameter. For PatchGCN [5] and TransMIL [36], the hyper-parameters were set exactly as in the original papers, except for the inserted PAMoE module. For LongViT [39], we set the number of transformer layers to 4, the hidden dimension to 512, the dilated attention ratio to (1, 2, 4, 8, 16, 32), and the segment length to (1024, 2048, 4096, 8192, 16384, 32768). We use global attention pooling as CLAM [29] to gather all tokens to the WSI-level feature. Unless otherwise specified, for all PAMoE layers, we set Prior Supervised Experts to 4, the number of Free Experts to 2, the capacity factor $c = 2.0$, and the tuning hyperparameter α of $\mathcal{L}_{\text{PAMoE}}$ is 0.1.

Training and Evaluation. Adam optimization [23] is adopted to optimize our model. We use Adam optimization with a default learning rate of 2×10^{-4}, weight decay of 1×10^{-5}, and the batch size is set to 6. All experiment results are obtained through 5-fold cross-validation. Concordance index (C-index) [15] and its standard deviation (std) are used to measure the predictive performance in correctly ranking the survival risk of each patient. All the experiments are implemented using PyTorch [33] on a workstation with an A6000 GPU.

11. Further Ablation

11.1. Ablation Study on Expert Choice Routing.

We introduced an expert choice routing strategy within PAMoE, which grants the module the capability to actively discard irrelevant patches. In this ablation study, we evaluate the impact of the PAMoE module relative to the vanilla SwitchMoE [8] on survival prediction outcomes. For SwitchMoE, we set the hyperparameters to match those of PAMoE, with a total of 6 experts and a capacity factor of 2.0. These experiments were performed using TransMIL, with the results are detailed in Tab. 5. The findings demonstrate that PAMoE consistently achieves superior performance over SwitchMoE across most datasets. Additionally, the performance of SwitchMoE underscores the general effectiveness of mixture-of-experts modules in addressing heterogeneous pathological tissues.

Table 5. Ablation study of the PAMoE and SwitchMoE. The best results over all models are highlighted in **bold**.

	COAD	LGG	LUAD	PAAD	BRCA
TransMIL	0.676 ± 0.028	0.770 ± 0.059	0.656 ± 0.033	0.637 ± 0.072	0.692 ± 0.049
TransMIL+SwitchMoE	0.679 ± 0.023	0.772 ± 0.068	0.666 ± 0.014	0.648 ± 0.073	0.669 ± 0.039
TransMIL+PAMoE	$\mathbf{0.688 \pm 0.036}$	$\mathbf{0.779 \pm 0.072}$	$\mathbf{0.668 \pm 0.016}$	$\mathbf{0.655 \pm 0.074}$	$\mathbf{0.694 \pm 0.054}$

Table 6. Ablation study of Number of the class token and residual connections. The *ffn_cls* column indicates whether an additional MLP layer is used for processing the class token, and the *residual* column specifies whether residual connections were applied.

ffn_cls	residual	COAD	LGG	LUAD	PAAD	BRCA
✗	✓	0.682 ± 0.032	$\mathbf{0.782 \pm 0.072}$	0.666 ± 0.020	0.647 ± 0.087	0.692 ± 0.057
✓	✓	0.687 ± 0.024	0.776 ± 0.056	0.665 ± 0.016	0.647 ± 0.084	0.681 ± 0.046
✓	✗	0.688 ± 0.036	0.779 ± 0.072	$\mathbf{0.668 \pm 0.016}$	$\mathbf{0.655 \pm 0.074}$	$\mathbf{0.694 \pm 0.054}$
✗	✗	$\mathbf{0.697 \pm 0.036}$	0.779 ± 0.061	0.649 ± 0.030	0.654 ± 0.070	0.678 ± 0.038

11.2. Discussions of Integrating PAMoE with Transformer-Based Methods

Transformer-based models typically use a **class token** to aggregate global information and apply **residual connections** in the feed-forward network (FFN) layer. When integrating such models with PAMoE, due to the instance drop characteristic of Mixture-of-Experts via Expert Choice Routing, the following issues may arise: For the class token, it might be treated as an unimportant input and discarded. For residual connections, if they are used, the dropped instances will not be dropped out but rather remain unprocessed. Therefore, in this section, we discuss the handling of class tokens and residual connections when applying PAMoE to transformer-based models. For **class tokens**, we experimented with two approaches: a) Treating the class token as a regular input without special handling; b) Excluding the class token from the PAMoE input and instead using an additional MLP layer for processing it. For **residual connections**, we evaluated two scenarios: a) Using residual connections as in the original architecture; b) Omitting residual connections.

Tab. 6 presents the performance of the TransMIL-based model under different configurations. The *ffn_cls* column indicates whether an additional MLP layer is used for processing the class token, and the *residual* column specifies whether residual connections were applied. We find that for residual connections, models without residual connections often perform better than those with residual connections. For the class token, when residual connections are used, the additional MLP layer for the class token has little impact on the results. However, if residual connections are not used and the additional MLP layer for the class token is also omitted, the model's performance shows a noticeable decline. We believe that the instance drop feature provides a performance boost for PAMoE. However, using residual connections may introduce a chaotic mapping space that confuses the model. For the class token, PAMoE's router might drop the class token, significantly impacting the final results. When residual connections are set, the class token is not entirely dropped, so the impact is smaller. Conversely, without residual connections, the class token is directly dropped, greatly affecting the model's performance. Therefore, in our implementation, we use a setting with an additional MLP layer for the class token and omit residual connections.

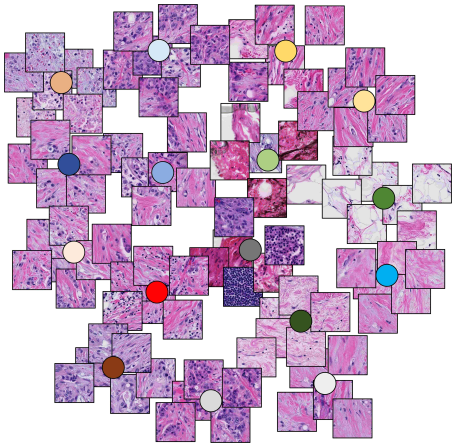

Figure 7. The visualization of the 16 clustering centers from K-means. We selected the patches with the highest cosine similarity to the clustering centers to represent each category.

11.3. Ablation Study on the Source of Prototypes

Although calculating prototypes based on classification results can provide supervision for expert preferences based on pathological priors, using an extra classifier introduces additional computational overhead. Therefore, we discuss an alternative approach that directly computes prior prototypes

Table 7. Ablation study of the source of prototypes.

	COAD	LGG	LUAD	PAAD	BRCA
TransMIL	0.676 ± 0.028	0.770 ± 0.062	0.644 ± 0.026	0.646 ± 0.026	0.692 ± 0.049
Classifier(PAMoE)	0.688 ± 0.036	0.779 ± 0.072	0.668 ± 0.016	0.655 ± 0.074	0.694 ± 0.054
K-means(n=4)	0.680 ± 0.047	0.785 ± 0.068	0.657 ± 0.030	0.667 ± 0.064	0.690 ± 0.052
Selected 4 from K-means(n=16)	0.682 ± 0.045	0.792 ± 0.060	0.665 ± 0.014	0.667 ± 0.079	0.698 ± 0.051

Table 8. Experiment of using CONCH as encoder.

	COAD	LGG	LUAD	PAAD	BRCA
TransMIL	0.700 ± 0.050	0.769 ± 0.026	0.623 ± 0.053	0.652 ± 0.063	0.676 ± 0.072
TransMIL+PAMoE	$\mathbf{0.706 \pm 0.053}$	$\mathbf{0.776 \pm 0.034}$	$\mathbf{0.648 \pm 0.060}$	$\mathbf{0.657 \pm 0.060}$	$\mathbf{0.682 \pm 0.059}$
LongViT	0.653 ± 0.042	0.721 ± 0.042	0.626 ± 0.042	0.601 ± 0.049	0.638 ± 0.049
TransMIL+PAMoE	$\mathbf{0.677 \pm 0.034}$	$\mathbf{0.766 \pm 0.061}$	$\mathbf{0.640 \pm 0.054}$	$\mathbf{0.629 \pm 0.040}$	$\mathbf{0.654 \pm 0.026}$

using cluster centers as supervisory targets here.

The results of the ablation study are shown in Tab. 7. We first attempted to directly cluster all patches using the K-means method (n=4) and used the resulting four cluster centers as supervisory target prototypes, which is shown in the *K-means (n=4)* row in the table. Next, we attempted to select tissue categories aligned with our priors from multiple clustering centers to achieve the same goal of incorporating pathological priors, which is shown in the *Selected 4 from K-means(n=16)* row. Specifically, we first applied K-means clustering (n=16) to the patch set, obtaining 16 clustering centers. For each clustering center, we selected the patches with the highest cosine similarity to the clustering centers for qualitative analysis, as shown in Fig. 7. We relied on pathologists to guide us in selecting clustering centers that represent the four categories in Eq. (11). Additionally, we attempt to use a classifier to perform majority voting among the top k patches with the highest cosine similarity to each clustering center, selecting prior-aligned clustering centers. This approach yielded results consistent with the pathologists' selections. The results indicate that the approach directly using four clustering centers from K-means shows a performance drop compared to the supervised methods, but it still outperforms the unsupervised baseline. The selection-based approach demonstrates more stable performance and even surpasses the classifier-based method on some datasets, showcasing its effectiveness. Therefore, we propose it as an alternative to the classifier-based prototype extraction method.

11.4. Discussion about Using CONCH as encoder

In experiments, CONCH was used as an additional classifier to obtain prototypes, which may lead to additional computational overhead and unfair comparisons. To address this concern, we conduct experiments of using CONCH as the encoder, and directly use CONCH text encoder to obtain features from tissue text prompts as prototypes. Tab. 8 shows that PAMoE can still bring improvements when CONCH is used as the encoder. Using CONCH as the encoder and leveraging the CONCH text encoder to extract prototypes can significantly improve efficiency. However, this requires using features after the projection head and normalization to align the text encoder and image encoder features, which is inconsistent with CONCH's setting when used as an image encoder, where image embeddings are usually taken before the projection head and normalization. Moreover, relying solely on CONCH's image and text encoders as feature extractors limits the scalability of PAMoE. Therefore, the primary method in this study employs CONCH only as a classifier to obtain prior prototypes, with additional experiments provided to explore the use of CONCH as an encoder.

11.5. Capacity Factor

The capacity factor c determines the number of instances each expert in PAMoE can process, directly influencing the proportion of instances dropped by the model. This instance drop characteristic makes PAMoE more sensitive to the setting of the capacity factor. Fig. 8 illustrates the performance of the TransMIL-based model under different capacity factor settings. We observed that under lower capacity factor settings, the model's performance on the COAD and BRCA datasets was significantly lower compared to other settings. However, for the LGG and PAAD datasets, lower capacity factor settings led to performance improvements. We believe this is related to the WSI resolution corresponding to the cancer type datasets. The average number of patches obtained from each dataset at $20\times$ magnification is as follows: COAD

Table 9. Proportion of discarded patches in different capacity factor settings and different datasets (%).

Capacity Factor	COAD	LGG	LUAD	PAAD	BRCA
2	2.95 ± 2.86	4.90 ± 2.77	5.17 ± 3.07	0.99 ± 0.50	5.13 ± 4.14
1.9	4.16 ± 3.68	6.68 ± 3.36	6.92 ± 3.63	1.67 ± 0.70	6.78 ± 4.83
1.8	5.64 ± 4.43	8.73 ± 3.96	8.97 ± 4.18	2.60 ± 0.95	8.70 ± 5.47
1.5	11.89 ± 6.38	16.68 ± 5.49	16.83 ± 5.42	7.54 ± 1.87	16.03 ± 6.91
1.4	14.68 ± 6.85	19.93 ± 5.86	20.05 ± 5.66	10.07 ± 2.17	19.03 ± 7.17
1	29.71 ± 7.12	35.86 ± 6.23	35.54 ± 5.53	25.09 ± 2.96	33.90 ± 6.88
0.9	34.53 ± 6.76	40.60 ± 6.00	40.11 ± 5.22	30.18 ± 2.97	38.41 ± 6.45

(8,091), LGG (10,122), LUAD (11,755), PAAD (12,356), and BRCA (9,633). Among them, the average number of patches in the COAD and BRCA is significantly lower than in other datasets. The capacity factor affects the proportion of patches discarded by PAMoE. With a smaller capacity factor setting, each expert processes fewer patches, leading to the discarding of more patches. For large-scale WSIs, the model may struggle with excessive input, so discarding more low-relevance patches might help the model focus on high-relevance patches, thereby improving overall performance. However, for smaller-scale WSIs, the model might already be able to focus on the patches by itself. In this case, excessive discarding could lead to information loss, thereby impacting the overall performance of the model.

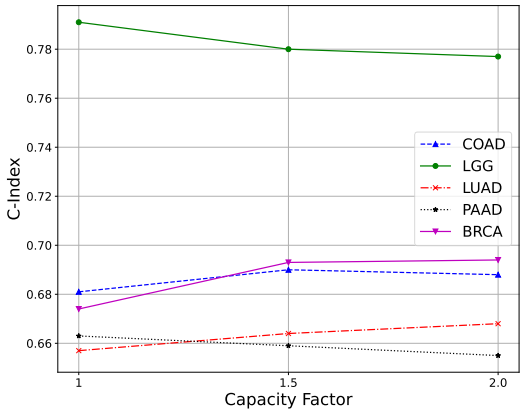

Figure 8. The impact of the capacity factor on model performance based on TransMIL.

11.6. Quantitative Analysis of Discarded Patches and Discussion about Capacity Factor

Tab. 9 presents the proportion of dropped patches of one PAMoE layer under different capacity factor settings and different datasets. The observations indicate that the proportion of dropped patches varies even under the same capacity factor setting. The classical capacity factor cannot directly reflect the patch dropping ratio in MoE via expert choice routing and is susceptible to factors such as input scale and the number of experts. In future work, we will explore more suitable methods to control the patch selection process in the MoE layer.

11.7. Further Discussion about Free Experts

Tab. 2 in the main paper presents the model performance under different experts number settings. The observations reveal that the setting without any free experts achieves the best results on certain datasets. This raises concerns about the necessity of free experts. We think the optimal number of free experts might have something to do with the task. If the predefined prototypes revel most of information required by the task, the network may not need free experts at all. But if the predefined prototypes can not differentiate classes required for the task, e.g. fine-grained classes, we may need to lean some free experts. We appreciate all reviewers' suggestions and will further explore it in future work.

12. Experiments on More Tasks

In the main paper, we primarily conducted detailed experiments based on the survival prediction task. To validate the generalizability of PAMoE across a broader range of tasks, we additionally evaluate the performance of PAMoE on **subtyping** and **staging** tasks. All settings are identical to those in the main paper.

12.1. Datasets.

Subtyping. We validated PAMoE on the subtyping task across two cancer types, including: COAD: Mucinous adenocarcinoma or not (2 classes, 434 cases); BRCA: Lobular carcinoma and Infiltrating duct carcinoma (2 classes, 950 cases),

Staging. We validated PAMoE on the staging task across four cancer types, including COAD (400 cases), LUAD (492 cases), PAAD (192 cases), and BRCA (999 cases). All the cases are divided into the "Stage I", "Stage II", "Stage III", and "Stage IV" classes. We choose these cancer types

Table 10. Subtyping results over two cancer datasets based on accuracy, F1, and AUC (mean ± std). The optimal results for different variants of the model are highlighted in **bold**.

	COAD			BRCA		
	Accuracy	F1	AUC	Accuracy	F1	AUC
PatchGCN	$\mathbf{0.882 \pm 0.043}$	$\mathbf{0.928 \pm 0.029}$	0.822 ± 0.028	$\mathbf{0.925 \pm 0.032}$	$\mathbf{0.951 \pm 0.023}$	0.907 ± 0.037
PatchGCN+PAMoE	0.847 ± 0.038	0.906 ± 0.024	$\mathbf{0.826 \pm 0.035}$	0.918 ± 0.025	0.947 ± 0.017	$\mathbf{0.923 \pm 0.026}$
TransMIL	0.858 ± 0.031	0.916 ± 0.020	$\mathbf{0.751 \pm 0.040}$	$\mathbf{0.941 \pm 0.013}$	0.956 ± 0.013	0.924 ± 0.022
TransMIL+PAMoE	$\mathbf{0.908 \pm 0.008}$	$\mathbf{0.947 \pm 0.006}$	0.745 ± 0.069	0.933 ± 0.018	$\mathbf{0.962 \pm 0.010}$	$\mathbf{0.928 \pm 0.013}$
LongViT	$\mathbf{0.707 \pm 0.285}$	0.709 ± 0.342	$\mathbf{0.503 \pm 0.017}$	0.806 ± 0.053	0.891 ± 0.032	0.543 ± 0.087
LongViT+PAMoE	0.687 ± 0.266	$\mathbf{0.734 \pm 0.367}$	0.499 ± 0.085	$\mathbf{0.812 \pm 0.031}$	$\mathbf{0.908 \pm 0.032}$	$\mathbf{0.554 \pm 0.079}$

Table 11. Staging results over four cancer datasets based on accuracy, F1, and AUC (mean ± std). The optimal results for different variants of the model are highlighted in **bold**.

	COAD			LUAD		
	Accuracy	F1	AUC	Accuracy	F1	AUC
PatchGCN	0.392 ± 0.014	0.382 ± 0.015	0.636 ± 0.029	0.549 ± 0.045	$\mathbf{0.439 \pm 0.049}$	0.678 ± 0.051
PatchGCN+PAMoE	$\mathbf{0.407 \pm 0.023}$	$\mathbf{0.389 \pm 0.035}$	$\mathbf{0.642 \pm 0.044}$	$\mathbf{0.584 \pm 0.062}$	0.432 ± 0.041	$\mathbf{0.687 \pm 0.033}$
TransMIL	0.426 ± 0.031	0.393 ± 0.032	0.649 ± 0.015	0.549 ± 0.047	$\mathbf{0.467 \pm 0.035}$	0.658 ± 0.062
TransMIL+PAMoE	$\mathbf{0.433 \pm 0.057}$	$\mathbf{0.406 \pm 0.054}$	$\mathbf{0.650 \pm 0.033}$	$\mathbf{0.569 \pm 0.039}$	0.429 ± 0.029	$\mathbf{0.681 \pm 0.059}$
LongViT	0.317 ± 0.109	0.135 ± 0.055	$\mathbf{0.508 \pm 0.033}$	0.471 ± 0.110	0.192 ± 0.016	0.518 ± 0.054
LongViT+PAMoE	$\mathbf{0.371 \pm 0.058}$	$\mathbf{0.160 \pm 0.044}$	0.483 ± 0.043	$\mathbf{0.537 \pm 0.061}$	$\mathbf{0.217 \pm 0.040}$	$\mathbf{0.523 \pm 0.039}$
	PAAD			BRCA		
	Accuracy	F1	AUC	Accuracy	F1	AUC
PatchGCN	0.802 ± 0.126	$\mathbf{0.527 \pm 0.093}$	0.623 ± 0.047	$\mathbf{0.526 \pm 0.053}$	0.386 ± 0.026	0.668 ± 0.030
PatchGCN+PAMoE	$\mathbf{0.868 \pm 0.057}$	0.480 ± 0.042	$\mathbf{0.669 \pm 0.053}$	0.514 ± 0.039	$\mathbf{0.397 \pm 0.033}$	$\mathbf{0.676 \pm 0.033}$
TransMIL	0.855 ± 0.055	0.450 ± 0.065	0.602 ± 0.052	0.499 ± 0.045	0.365 ± 0.024	0.651 ± 0.035
TransMIL+PAMoE	$\mathbf{0.873 \pm 0.048}$	$\mathbf{0.514 \pm 0.058}$	$\mathbf{0.629 \pm 0.069}$	$\mathbf{0.500 \pm 0.039}$	$\mathbf{0.372 \pm 0.031}$	$\mathbf{0.654 \pm 0.031}$
LongViT	$\mathbf{0.813 \pm 0.082}$	0.346 ± 0.077	0.478 ± 0.066	$\mathbf{0.576 \pm 0.033}$	0.187 ± 0.009	0.508 ± 0.036
LongViT+PAMoE	0.710 ± 0.260	$\mathbf{0.347 \pm 0.127}$	$\mathbf{0.579 \pm 0.045}$	0.554 ± 0.042	$\mathbf{0.316 \pm 0.018}$	$\mathbf{0.642 \pm 0.044}$

for training and evaluation using the following criteria: 1) overall label available, 2) balanced distribution of subtyping and staging labels. We use accuracy, F1, and AUC metrics for evaluation.

12.2. Results

The results for the staging and subtyping tasks are respectively presented in Tab. 11 and Tab. 10. We observe that the findings for these tasks are consistent with the conclusions drawn from the survival prediction task in Sec. 5.3 in main paper, demonstrating the generalizability of PAMoE. Additionally, we find that for staging and subtyping tasks, PAMoE exhibit relatively stable improvements when applied to the PatchGCN model, which is not transformer-based. This suggests the potential of PAMoE for applications in non-transformer architectures. We will further explore this in future research.