Shazam: Unifying Multiple Foundation Models for Advanced Computational Pathology

Wenhui Lei[1,2,†], Anqi Li[3,†], Yusheng Tan[3,†], Hanyu Chen[4], and Xiaofan Zhang[1,2,✉]

[1] Shanghai Jiaotong University
[2] Shanghai Artificial Intelligence Laboratory
[3] Washington University in St. Louis
[4] The First Hospital of China Medical University
xiaofan.zhang@sjtu.edu.cn

Abstract. Foundation Models (FMs) in computational pathology (CPath) have significantly advanced the extraction of meaningful features from histopathology image datasets, achieving strong performance across various clinical tasks. Despite their impressive performance, these models often exhibit variability when applied to different tasks, prompting the need for a unified framework capable of consistently excelling across various applications. In this work, we propose Shazam, a novel framework designed to efficiently combine multiple CPath models. Unlike previous approaches that train a fixed-parameter FM, Shazam dynamically extracts and refines information from diverse FMs for each specific task. To ensure that each FM contributes effectively without dominance, a novel distillation strategy is applied, guiding the student model with features from all teacher models, which enhances its generalization ability. Experimental results on two pathology patch classification datasets demonstrate that Shazam outperforms existing CPath models and other fusion methods. Its lightweight, flexible design makes it a promising solution for improving CPath analysis in real-world settings. Code will be available at here.

Keywords: Foundation Model · Computational Pathology · Knowledge Distillation.

1 Introduction

Recent advances in FMs for CPath have dramatically transformed the field by enabling the extraction of robust representations from extensive collections of histopathology images [2,3,4,8,12,13]. Often trained through self-supervised learning, these models have achieved remarkable performance across a diverse range of clinical tasks. However, their effectiveness can vary depending on the specific application, raising a critical question: Can we develop a unified model

† Contributed equally to this work.

that integrates multiple FMs to consistently deliver superior performance across various downstream CPath tasks?

GPFM [9] is the first attempt to integrate knowledge from multiple CPath FMs. It employs a knowledge distillation strategy [5,7] to build an FM that leverages the collective knowledge of several teacher models. However, GPFM's primary goal is to create a static, fixed-parameter model. This approach presents challenges in the fast-evolving CPath field, where maintaining high performance across continuously emerging models becomes difficult due to the significant retraining costs. In contrast, our work proposes a lightweight, efficient solution capable of adapting quickly to various tasks or FM combinations with minimal computational overhead.

To address this challenge, we propose **Shazam**, a novel framework designed to integrate multiple FMs for CPath analysis. As illustrated in Fig. 1, the features extracted by the teacher FMs are stacked and used as input to the student model, enabling it to receive diverse feature representations. The student network, composed of self-attention layers, performs feature fusion across the different teacher models. To prevent any single teacher model from dominating and potentially impairing the student model's generalization ability, we employ a distillation strategy, where the final feature representation of the student is supervised by the features from all teacher models. Extensive experiments on two pathology patch classification datasets demonstrate that Shazam significantly outperforms the CPath FMs it is based on, as well as other feature fusion methods. The lightweight, flexible, and highly effective nature of Shazam makes it a valuable tool for enhancing the performance of CPath analysis in real-world applications.

2 Method

2.1 Internal Structure of Shazam

The overall architecture of Shazam is depicted in Figure 1. It mainly contains two principal components: (1) **Teacher Model**, which extracts diverse feature representations from the input pathology patches, and (2) **Student Model**, which learns from teacher representations through the integration of a self-attention mechanism and a classification module.

Teacher Model: Given pathology patches, we employ N pre-trained teacher models to extract feature representations. Let $F_i \in \mathbb{R}^{D_i}$ denote the extracted feature from the i-th teacher model, where D_i represents the original feature dimension of the teacher model. To ensure consistency across different teacher outputs, we project each extracted feature into a D-dimensional space $T_i \in \mathbb{R}^D$, using a learnable linear transformation:

$$T_i = W_i F_i + b_i \quad i = 1, \ldots, N, \tag{1}$$

where $W_i \in \mathbb{R}^{D \times D_i}$ and $b_i \in \mathbb{R}^D$ are learnable parameters for each teacher model. Then we stack these embeddings vertically to form a structured representation $T = [T_1, T_2, \ldots, T_N] \in \mathbb{R}^{N \times D}$.

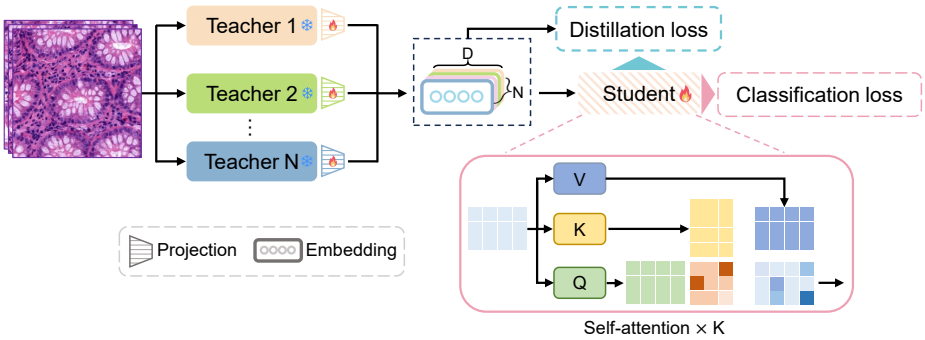

Fig. 1. Overview of the Proposed Framework, Shazam. In the feature extraction stage, multiple teacher models extract feature representations from the input data, which are then projected and stacked as input to the student module. The student module, equipped with K self-attention layers, refines these representations for downstream tasks. During training, classification and distillation losses are aggregated to guide the student model, ensuring effective knowledge transfer while preserving classification performance.

Student Model: Our student model, *Shazam*, aims to replicate the collective knowledge of the teacher models in a more compact form. It consists of multiple self-attention layers followed by a classifier.

For each self-attention layer, we define the query Q, key K, and value V matrices as transformations of the current embeddings T:

$$Q = W_Q T, \quad K = W_K T, \quad V = W_V T, \quad Q, K, V \in \mathbb{R}^{N \times D}, \tag{2}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{D \times D}$ are learnable projection matrices that transform the current features into a unified space for attention computation. The attention mechanism is then applied:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{D}}\right) V, \tag{3}$$

where the attention score matrix $\frac{QK^\top}{\sqrt{D}}$ captures the relationships between different teacher embeddings. After processing through K self-attention layers, we obtain the fused representation $T^* \in \mathbb{R}^{N \times D}$, where T_i^* $(i = 1, \ldots, N)$ corresponds to the i-th embedding in T^*. We then perform an averaging operation along the dimension N to aggregate the information:

$$T_{\text{final}} = \frac{1}{N} \sum_{i=1}^{N} T_i^*, \quad T_{\text{final}} \in \mathbb{R}^D. \tag{4}$$

Then T_{final} is passed through a multilayer perceptron (MLP) classifier, producing the logits $Z \in \mathbb{R}^C$, where C denotes the number of classes in the dataset.

2.2 Distillation Methodology

To optimize parameters for the specific task and regularize for the teacher models, we utilize the final embedding T_{final} from multiple attention layers as the student's representation and distill knowledge from teacher models. Our distillation process consists of two key components: Cosine Similarity Loss ($\mathcal{L}_{Cosine}$) and Huber Loss ($\mathcal{L}_{\text{Huber}}$).

Cosine Similarity Loss: To preserve the relative geometric structure of the teacher features, which is crucial for maintaining the discriminative relationships in the learned representation space, we employ a cosine similarity loss. It is computed for each teacher model to ensure that the overall direction of the student's feature representation aligns with that of each teacher. The cosine similarity loss between T_{final} and T_i is defined as

$$\mathcal{L}_{\text{Cosine}} = \sum_{i=1}^{N} \left(1 - \frac{\langle T_{\text{final}}, T_i \rangle}{\|T_{\text{final}}\| \cdot \|T_i\|}\right). \tag{5}$$

This formulation emphasizes the overall alignment by averaging the angular differences across all teachers, thereby encouraging the student model to capture the general directional trend of the teacher representations.

Huber Loss: To mitigate the influence of noisy teacher features and ensure that the student learns consistent feature magnitudes, we employ the Huber Loss. This loss is applied independently to each teacher feature, effectively reducing the impact of extreme values in any specific channel or teacher. As a result, the student model is less likely to overfit such anomalies. The overall Huber loss is computed by summing the individual losses overall N teacher models:

$$\mathcal{L}_{\text{Huber}} = \sum_{i=1}^{N} HuberLoss(T_{\text{final}}, T_i). \tag{6}$$

The Huber loss for a single pair of features is defined as

$$HuberLoss(T_{\text{final}}, T_i) = \begin{cases} \frac{1}{2}\|T_{\text{final}} - T_i\|^2, & \text{if } \|T_{\text{final}} - T_i\| \leq \delta, \\ \delta\|T_{\text{final}} - T_i\| - \frac{1}{2}\delta^2, & \text{if } \|T_{\text{final}} - T_i\| > \delta, \end{cases} \tag{7}$$

where the threshold parameter δ determines the point at which the loss function transitions from a quadratic to a linear penalty. This formulation preserves the smooth quadratic behavior for small discrepancies, facilitating stable optimization, while applying a linear penalty for larger differences to improve robustness against outliers.

2.3 Overall Objective Function

Building upon the student module and the distillation methodology, we further designed the objective function for model training. For classification tasks, the

Cross Entropy Loss ($\mathcal{L}_{CE}$) is adopted, which is defined as:

$$\mathcal{L}_{CE} = -\sum_{c=1}^{C} y_{i,c} \log \left(\frac{e^{z_{i,c}}}{\sum_{c'=1}^{C} e^{z_{i,c'}}} \right), \tag{8}$$

where C represents the number of classes, $y_{i,c}$ is the ground truth label for the i-th sample belonging to class c, and $z_{i,c}$ denotes the predicted logit.

We combine the Cross Entropy Loss with the proposed distillation methods and introduce a hyperparameter λ to balance the contributions from the classification and distillation losses. The overall objective function $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \left(\mathcal{L}_{Cosine} + \mathcal{L}_{Huber} \right). \tag{9}$$

3 Experiments and Results

3.1 Dataset and Descriptions

HunCRC for CRC Tissue Classification (9 Classes): This dataset comprises 101,398 H&E-stained image patches, also referred to as regions of interest (ROIs), each measuring 512 × 512 pixels at a resolution of 0.48 microns per pixel (mpp). These ROIs were extracted from 200 formalin-fixed paraffin-embedded (FFPE) whole-slide images (WSIs) of colorectal biopsies[10]. Designed to support the training of machine learning models for colorectal tissue classification, the dataset includes annotations for nine classes: Adenocarcinoma (4,315 ROIs), High-Grade Dysplasia (2,281 ROIs), Low-Grade Dysplasia (55,787 ROIs), Inflammation (763 ROIs), Tumor Necrosis (365 ROIs), Suspicious for Invasion (570 ROIs), Resection Edge (534 ROIs), Technical Artifacts (3,470 ROIs), and Normal Tissue (31,323 ROIs).

For training and evaluation, the dataset follows the official split, with 76,753 ROIs used for training, 11,327 ROIs for validation, and 11,328 ROIs for testing.

UniToPatho for CRC Polyp Classification (6 Classes): This dataset comprises 9,536 H&E-stained image patches (ROIs) extracted from 292 whole-slide images (WSIs)[1]. Its primary purpose is to facilitate the training of deep neural networks for colorectal polyp classification and adenoma grading. The dataset includes annotations for six classes: Normal Tissue (950 ROIs), Hyperplastic Polyp (545 ROIs), Tubular Adenoma with High-Grade Dysplasia (454 ROIs), Tubular Adenoma with Low-Grade Dysplasia (3,618 ROIs), Tubulo-Villous Adenoma with High-Grade Dysplasia (916 ROIs), and Tubulo-Villous Adenoma with Low-Grade Dysplasia (2,186 ROIs).

For training and evaluation, the dataset follows the official split, with 6,270 ROIs used for training, 1,199 ROIs for validation, and 1,200 ROIs for testing.

3.2 Implementation Details

We employ **UNI[3], Phikon-v2[4], Virchow[12]** and **GigaPath[13]** as our teacher models. For the classification task, the features extracted from each

Model	UniToPatho			HunCRC		
	Balanced Acc	Weighted F1	Top-1 Acc	Balanced Acc	Weighted F1	Top-1 Acc
Virchow	0.484	0.512	0.529	0.318	0.759	0.752
- fine-tune	0.490	0.513	0.547	0.402	0.808	0.804
UNI	0.507	0.514	0.533	0.381	0.788	0.788
- fine-tune	0.458	0.510	0.540	0.439	0.807	0.808
PhiKon	0.492	0.512	0.544	0.406	0.801	0.799
- fine-tune	0.459	0.477	0.522	0.392	0.803	0.802
GigaPath	0.467	0.500	0.520	0.432	0.815	0.811
- fine-tune	0.434	0.452	0.494	0.448	0.821	0.823
Shazam	**0.551**	**0.587**	**0.598**	**0.517**	**0.846**	**0.842**

Table 1. Performance comparison among Shazam, the four teacher models, and their fine-tuned counterparts on UniToPatho and HunCRC datasets.

teacher model are fed into an MLP classifier for prediction. Additionally, each teacher model is fine-tuned separately on two datasets, and the output features are subsequently classified using the same method. After obtaining the classification results, we compute the relevant performance metrics for model comparison.

The training and inference of all experiments were conducted on an NVIDIA RTX 4090 GPU with a batch size of 64. The training process was conducted for 100 epochs, taking 1 hour for UniToPatho and 8 hours for HunCRC. All models were optimized using AdamW with an initial learning rate of 1e-3, and the weight decay was set to 1e-3. The threshold δ of Huber Loss was set to 1.0. The hyperparameter λ of the objective loss function was set to 0.5 for the UniToPatho dataset and 0.05 for the HunCRC dataset. Evaluation was performed using the Balanced Accuracy (BA), Weighted F1-score, and Top-1 Accuracy metrics.

3.3 Comparison Results

We fine-tune all four teacher models and compare our method with both the original (frozen) and fine-tuned (trainable) versions of these teacher models across each dataset. In both cases, an MLP classifier is added, trained with a learning rate of 1e-3. The fine-tuned models also update the backbone with a lower learning rate of 1e-5. Furthermore, we evaluate Shazam in relation to other methods. The experimental results are reported in Table 1 and Figure 2. Our method achieves the best performance across every metric, especially on UniToPatho dataset, where our method outperforms other methods by a large margin. The results demonstrate that our method effectively distills knowledge from teacher models to a student model, enhancing the student model's overall performance.

3.4 Ablation study

Student Structure: We compare Shazam with several widely used feature fusion methods, including *Sum*, *Concat*, *Mamba*[6], *MoE*[11], and *Shazam* without distillation. In the *Sum* method, teacher features are summed and averaged to form a unified feature, which is then passed to the classifier for prediction.

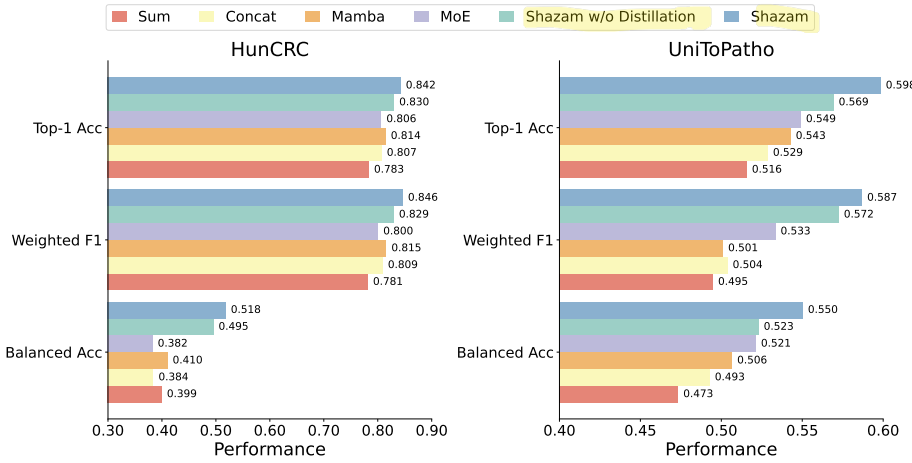

Fig. 2. Performance comparison of different feature fusion methods on UniToPatho and HunCRC datasets.

The *Concat* method concatenates teacher features before classification. For the *Mamba* method, we employ a Mamba model to fuse teacher features by first extracting deep representations with a Mamba block, then aggregating them via mean pooling, and finally applying an MLP classifier to generate predictions. For *MoE*, we utilize a mixture-of-experts strategy to integrate features. A gating network dynamically assigns weights to expert-projected features, which are then fused and processed by an MLP for classification. We evaluate both the *Shazam* framework and its variant without distillation against these methods, ensuring a fair and balanced comparison by removing the distillation mechanism where applicable. The experimental results are presented in Figure 2.

The results indicate that *Shazam* w/o distillation outperforms other methods on both datasets, demonstrating the superiority of our architecture. With distillation enabled, performance further improves, with Balanced Accuracy (BA) increasing from 0.495 to 0.518, Weighted F1-score from 0.829 to 0.846, and Top-1 Accuracy from 0.83 to 0.842 on HunCRC. Similarly, on UniToPatho, BA increases from 0.523 to 0.55, Weighted F1-score from 0.572 to 0.587, and Top-1 Accuracy from 0.569 to 0.598. These results validate the effectiveness of our design and highlight the additional benefits introduced by the distillation mechanism.

Layer Depth: To analyze the impact of the number of attention layers on Shazam's performance, we compare 1, 3, 5, and 7 attention layers on the same evaluation metrics — Balanced Accuracy, Weighted F1-score and Top-1 Accuracy. The experimental results are presented in Table 2. On the UniToPatho dataset, the model with 5 layers achieves the best performance across all metrics

Num Layers	UniToPatho			HunCRC		
	Balanced Acc	Weighted F1	Top-1 Acc	Balanced Acc	Weighted F1	Top-1 Acc
1-layer	**0.553**	0.569	0.573	0.483	0.834	0.835
3-layer	0.550	0.548	0.555	0.489	0.822	0.827
5-layer	0.551	**0.587**	**0.598**	**0.517**	**0.846**	**0.842**
7-layer	0.523	0.567	0.570	0.493	0.839	0.838

Table 2. Performance comparison across different numbers of layers for Shazam on UniToPatho and HunCRC datasets.

Value of λ	UniToPatho			HunCRC		
	Balanced Acc	Weighted F1	Top-1 Acc	Balanced Acc	Weighted F1	Top-1 Acc
1.00	**0.558**	0.556	0.571	0.466	0.825	0.825
0.50	0.551	**0.587**	**0.598**	0.486	0.812	0.813
0.10	0.530	0.563	0.565	0.497	0.830	0.831
0.05	0.540	0.543	0.563	**0.517**	**0.846**	**0.842**
0.01	0.518	0.562	0.580	0.509	0.820	0.816

Table 3. Performance comparison across different values of λ for distillation in Shazam on UniToPatho and HunCRC datasets.

except for Balanced Accuracy. Similarly, on the HunCRC dataset, the 5-layer model outperforms other configurations in all metrics.

These results confirm that a 5-layer attention structure offers the best trade-off between complexity and training efficiency, making it the optimal choice for both datasets.

Distillation Weight: To analyze the necessity of distillation, we evaluate the impact of various distillation weights λ. Specifically, we experiment with $\lambda =$ 1, 0.5, 0.1, 0.05, and 0.01 to determine the optimal balance between supervised learning and knowledge distillation. On the small-scale UniToPatho dataset, the best performance is obtained with $\lambda = 0.5$, whereas on the large-scale HunCRC dataset, the optimal result is achieved with $\lambda = 0.05$. The experimental results are presented in Table 3. Although distillation is critical for effective knowledge transfer, our findings indicate that for large-scale datasets, the supervised loss alone may provide sufficient learning signals, thereby allowing for a smaller distillation weight. Nevertheless, our method consistently achieves significant improvements irrespective of the dataset size.

4 Conclusion

In this work, we have introduced Shazam, a novel framework designed to integrate multiple foundation models (FMs) for computational pathology (CPath) analysis. By dynamically extracting and refining information from diverse teacher models through knowledge distillation, Shazam offers a lightweight, efficient solution that adapts flexibly to CPath tasks. Our extensive experiments demonstrate that Shazam significantly outperforms the compared CPath FMs and other fea-

ture fusion methods, highlighting its potential for improving CPath analysis in real-world applications. However, the analysis of whole slide image-level tasks is not addressed in this study, which will be explored in future work.

References

1. Barbano, C.A., Perlo, D., Tartaglione, E., Fiandrotti, A., Bertero, L., Cassoni, P., Grangetto, M.: Unitopatho, a labeled histopathological dataset for colorectal polyps classification and adenoma dysplasia grading. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 76–80. IEEE (2021)
2. Chen, K., Liu, M., Yan, F., Ma, L., Shi, X., Wang, L., Wang, X., Zhu, L., Wang, Z., Zhou, M., et al.: Cost-effective instruction learning for pathology vision and language analysis. arXiv preprint arXiv:2407.17734 (2024)
3. Chen, R.J., Ding, T., Lu, M.Y., Williamson, D.F., Jaume, G., Song, A.H., Chen, B., Zhang, A., Shao, D., Shaban, M., et al.: Towards a general-purpose foundation model for computational pathology. Nature Medicine **30**(3), 850–862 (2024)
4. Filiot, A., Jacob, P., Mac Kain, A., Saillard, C.: Phikon-v2, a large and public feature extractor for biomarker prediction. arXiv preprint arXiv:2409.09173 (2024)
5. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**(6), 1789–1819 (2021)
6. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
7. Hinton, G.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
8. Hua, S., Yan, F., Shen, T., Ma, L., Zhang, X.: Pathoduet: Foundation models for pathological slide analysis of h&e and ihc stains. Medical Image Analysis **97**, 103289 (2024)
9. Ma, J., Guo, Z., Zhou, F., Wang, Y., Xu, Y., Cai, Y., Zhu, Z., Jin, C., Lin, Y., Jiang, X., et al.: Towards a generalizable pathology foundation model via unified knowledge distillation. arXiv preprint arXiv:2407.18449 (2024)
10. Pataki, B.Á., Olar, A., Ribli, D., Pesti, A., Kontsek, E., Gyöngyösi, B., Bilecz, Á., Kovács, T., Kovács, K.A., Kramer, Z., et al.: Huncrc: annotated pathological slides to enhance deep learning applications in colorectal cancer screening. Scientific Data **9**(1), 370 (2022)
11. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
12. Vorontsov, E., Bozkurt, A., Casson, A., Shaikovski, G., Zelechowski, M., Severson, K., Zimmermann, E., Hall, J., Tenenholtz, N., Fusi, N., Yang, E., Mathieu, P., van Eck, A., Lee, D., Viret, J., Robert, E., Wang, Y.K., Kunz, J.D., Lee, M.C.H., Bernhard, J.H., Godrich, R.A., Oakley, G., Millar, E., Hanna, M., Wen, H., Retamero, J.A., Moye, W.A., Yousfi, R., Kanan, C., Klimstra, D.S., Rothrock, B., Liu, S., Fuchs, T.J.: A foundation model for clinical-grade computational pathology and rare cancers detection. Nature Medicine (2024)
13. Xu, H., Usuyama, N., Bagga, J., Zhang, S., Rao, R., Naumann, T., Wong, C., Gero, Z., González, J., Gu, Y., Xu, Y., Wei, M., Wang, W., Ma, S., Wei, F., Yang, J., Li, C., Gao, J., Rosemon, J., Bower, T., Lee, S., Weerasinghe, R., Wright, B.J., Robicsek, A., Piening, B., Bifulco, C., Wang, S., Poon, H.: A whole-slide foundation model for digital pathology from real-world data. Nature (2024)