PathVQ: Reforming Computational Pathology Foundation Model for Whole Slide Image Analysis via Vector Quantization

Honglin Li[1,2*] **Zhongyi Shui**[1,2*] **Yunlong Zhang**[1,2]
Chenglu Zhu[2†] **Lin Yang**[2,3,4†]
[1] College of Computer Science and Technology, Zhejiang University
[2] School of Engineering, Westlake University
[3] The Institute of Advanced Technology, Westlake Institute for Advanced Study
[4] Center for Interdisciplinary Research and Innovation, MuyuanLaboratory
{lihonglin,zhuchenglu,yanglin}@westlake.edu.cn

Abstract

Pathology whole slide image (WSI) analysis is vital for disease diagnosis and understanding. While foundation models (FMs) have driven recent advances, their scalability in pathology remains a key challenge. In particular, vision-language (VL) pathology FMs align visual features with language annotation for downstream tasks, but they rely heavily on large-scale image-text paired data, which is scarce thus limiting generalization. On the other hand, vision-only pathology FMs can leverage abundant unlabeled data via self-supervised learning (SSL). However, current approaches often use the [CLS] token from tile-level ViTs as slide-level input for efficiency (a tile with 224×224 pixels composed of 196 patches with 16×16 pixels). This SSL pretrained [CLS] token lacks alignment with downstream objectives, limiting effectiveness. We find that spatial patch tokens retain a wealth of informative features beneficial for downstream tasks, but utilizing all of them incurs up to 200× higher computation and storage costs compared [CLS] token only (e.g., 196 tokens per ViT_{224}). This highlights a fundamental trade-off between efficiency and representational richness to build scalable pathology FMs. To address this, we propose a feature distillation framework via vector-quantization (VQ) that compresses patch tokens into discrete indices and reconstructs them via a decoder, achieving 64× compression ($1024 \rightarrow 16$ dimensions) while preserving fidelity. We further introduce a multi-scale VQ (MSVQ) strategy, enhancing both reconstruction and providing SSL supervision for slide-level pretraining. Built upon MSVQ features and supervision signals, we design a progressive convolutional module and a slide-level SSL objective to learn spatially rich representations for downstream WSI tasks. Extensive experiments across multiple datasets demonstrate that our approach achieves state-of-the-art performance, offering a scalable and effective solution for high-performing pathology FMs in WSI analysis.

1 Introduction

Cancer remains one of the most challenging diseases to diagnose and prognosticate, with pathology playing a pivotal role in understanding its complexities [28]. Traditional histopathological analysis relies heavily on manual examination of tissue samples by pathologists, a process that is not only time-

*Equal Contribution
†Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

consuming but also prone to inter-observer variability [24]. In recent years, computational pathology has emerged as a transformative method, leveraging whole-slide images (WSIs) to enable automated and quantitative analysis of tissue samples [49, 79, 48]. WSIs, which are high-resolution digital scans of entire tissue slides, provide a wealth of information that can be harnessed for cancer diagnosis, prognosis, and treatment planning. However, the ultra-high resolution of WSIs, often exceeding billions of pixels, presents significant challenges for effective computational modeling [61, 41].

Recent advances in foundation models (FMs)[3, 6, 51, 70] have shown strong potential in computational pathology. Studies have demonstrated the effectiveness of self-supervised learning (SSL)[75, 10, 20, 52] and vision-language (VL) pretraining [17, 48, 57] in extracting semantic features on pathology images. The FMs typically process WSIs by dividing them into smaller tiles (e.g., 224×224 pixels as a tile), extracting features from each tile, and aggregating these features to make slide-level predictions. VL-FMs [57, 48] excel in downstream tasks (e.g. zero-/few-shot ROI classification [65]), but their scalability is limited by the scarcity of large-scale image-text pairs [29, 65, 30]. In contrast, vision-only FMs trained on unlabeled data via SSL are more scalable. However, most methods adopt the task-agnostic [CLS] token from pretrained ViTs as a global representation of each tile [48, 20] and fed as WSI input [31, 61, 10, 33, 40, 62, 43, 67]. This approach overlooks critical spatial information captured by other spatial tokens, which are particularly essential for modeling nuanced pathological variations in gigapixel WSIs.

Notably, some studies have attempted to address this issue by scaling up to larger models (UNI-2 [10] using ViT-giant) or combining [average pooling] features with the [CLS] token (Virchow-2 [90]). Disappointingly, these approaches yield only marginal improvements. Consequently, we argue that the scalability and performance of FMs is fundamentally constrained by the trade-off between efficiency (using [CLS] only) and representational richness (using all patch tokens): Leveraging all spatial patch tokens benefit WSI analysis but incurs nearly 200× higher storage and training costs as shown in Figure 1 (e.g., 196 tokens in ViT_{224}). To address this, we introduce feature distillation via vector quantization (VQ) [71, 53] on patch features, which efficiently compresses spatial patch tokens using discrete indices and a decoder. Our method reduces token dimensionality from 1024 to 16, achieving a 64× compression rate while preserving reconstruction fidelity. This compression process retains original spatial and contextual information, ensuring that critical features are preserved for downstream tasks.

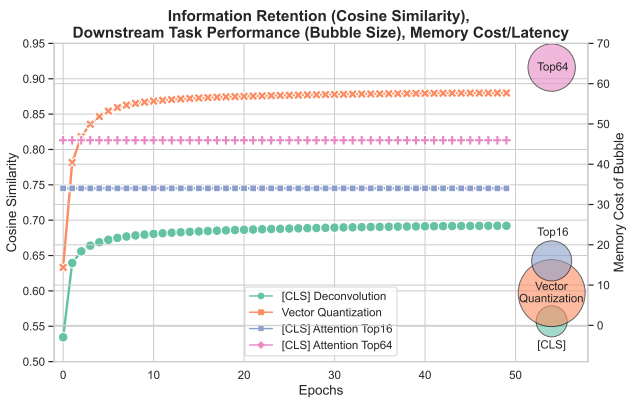

Figure 1: Evaluation on information loss via reconstruction training. Directly using the [CLS] token results in significant information loss, making it difficult to reconstruct all patch tokens, potentially discarding critical details for downstream tasks. In contrast, vector quantization retains more original information and show stronger result on downstream task 'BRACS'. The TopK patch tokens via CLS tokens attention selection are included for comparison.

Furthermore, we employ a multi-scale VQ (MSVQ) strategy, which unifies patch-level and tile-level feature VQ. Intuitively, tile-level feature like [CLS] token can be seen adaptive combination all patch features, thus they share the same feature space and can be learned into a single VQ model. The MSVQ not only enhances VQ reconstruction performance but also serves as a SSL supervision target for a seamless slide-level pretraining objective (working as a tokenizer thus can be pretrained like BERT [15, 25, 53]). By integrating slide SSL into our framework, we enable the model to learn rich, discriminative representations from unlabeled WSIs, addressing the challenge of limited WSI samples in computational pathology downstream tasks. Built upon the quantized features of patches and supervision targets of tiles via MSVQ, we develop a progressive convolutional module and slide-level SSL to extract representations with rich spatial information for downstream WSI tasks, leading to more accurate and interpretable predictions for

tasks like cancer diagnosis and prognosis. The contributions of our work can be summarized as follows:

1) Efficient Token Compression with VQ Distillation: We propose a novel VQ-based framework that compresses patch-level spatial tokens by 64× while retaining critical spatial and contextual information, enabling scalable and efficient WSI analysis.

2) SSL Supervision via offline tokenizer: Our improved MSVQ strategy not only enhances feature reconstruction but also serves as an SSL supervision target for slide-level mask prediction, providing a new direction for pretraining WSI models.

3) Rigorous Validation: Extensive evaluations on multiple datasets demonstrate the effectiveness of our approach, achieving state-of-the-art performance in WSI analysis tasks, with practical implications for clinical applications.

By addressing the computational challenges of WSI analysis while preserving critical spatial information, our framework offers a new perspective on the development of computational pathology foundation models, paving the way for more accurate and scalable cancer diagnostics.

2 Method

2.1 Preliminary

For WSI modeling, a WSI $\mathbf{X}$ is first divided into N tiles: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, which are then processed by the FM. The pretrained FM ViT converts tile image $\mathbf{x}$ into n patches $\mathbf{x} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]$, where the most commonly used patch size is 16×16. The ViT outputs all patch representations within a tile: $[\mathbf{s}; \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$, where $\mathbf{s}$ serves as a summary [CLS] of the spatial tokens of all patches ($\mathbf{ST} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$). Most existing approaches [31, 84, 10] rely on the [CLS] token from each tile to form WSI input embeddings $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{N \times D}$. These embeddings are subsequently aggregated for slide-level prediction: $\hat{Y} = g(\mathbf{S}; \theta)$, where $g(\theta)$ can be an attention [31] mechanism or a Transformer.

In contrast, this paper explores using all spatial tokens $\mathbf{H} = [\mathbf{ST}_1, \mathbf{ST}_2, \ldots, \mathbf{ST}_N] \in \mathbb{R}^{N \times n \times D}$ for slide-level prediction. Here, N (the number of tiles) can easily exceed 5k, $n = 196$ (the number of patches per tile), and the feature dimension $D = 1024$ (for UNI [10]). So, directly leveraging these high-dimensional data (about 1 million patch tokens) is computationally prohibitive for WSI training.

2.2 Vector Quantization Learning

To mitigate the computational burden while incorporating all patches' ST representations, we introduce vector-quantization (VQ) learning on the pretrained FM's patch ST, as illustrated in Figure 3b. This framework consists of an encoder, a quantizer, and a decoder. Additionally, we extend VQ to support both patch and tile representations via a multi-scale VQ strategy.

2.2.1 VQ for Patches

The spatial tokens ($\mathbf{ST}$) are mapped into discrete codes through vector quantization (VQ). Specifically, the tile-level representation $\mathbf{ST} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$ is first passed through an MLP encoder to reduce its dimensionality from D to d:

$$[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n] = \mathrm{Enc}([\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]), \tag{1}$$

where the resulting low-dimensional representations $[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n]$ are subsequently tokenized into discrete indices $\mathbf{ST}_{\mathrm{tok}} = [z_1, z_2, \ldots, z_n]$. The codebook $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_C] \in \mathbb{R}^{C \times d}$ consists of C learnable embeddings. Each patch-level representation $\mathbf{e}_i$ is assigned to its nearest neighbor in the codebook via:

$$z_i = \arg\min_j \left\| \ell_2(\mathbf{e}_i) - \ell_2(\mathbf{v}_j) \right\|_2, \tag{2}$$

where $j \in \{1, 2, \ldots, C\}$, and ℓ_2 denotes L_2 normalization used for distance computation, ensuring that each patch token is matched to the most similar codebook vector.

After quantization, the selected embeddings $\mathbf{E}_{z_1}, \mathbf{E}_{z_2}, \ldots, \mathbf{E}_{z_n}$ are passed to a multi-layer Transformer decoder to reconstruct the original spatial token representation. During training, the decoder output $\mathbf{o}_i$ is aligned with the target $\mathbf{h}_i$ by maximizing their cosine similarity.

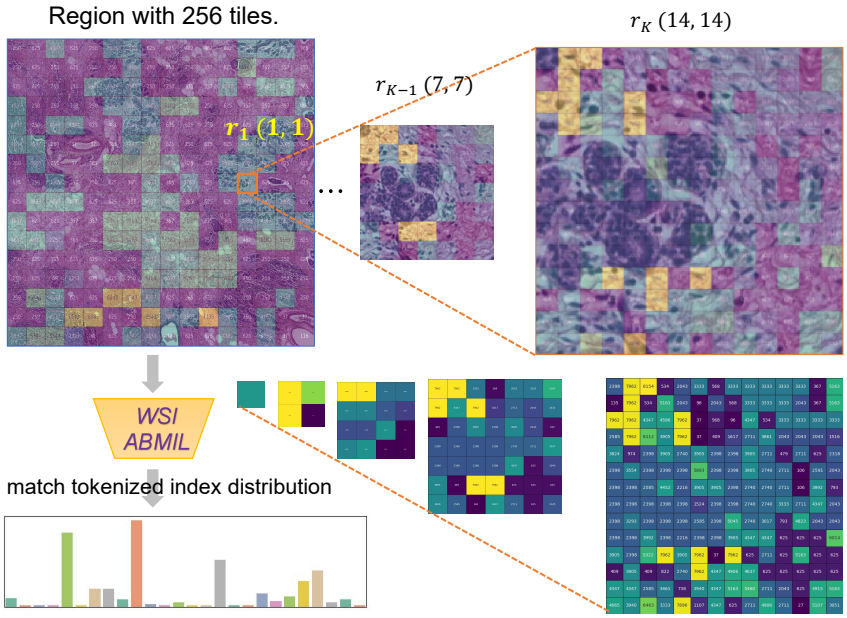

Figure 2: Multi-scale Vector Quantization (MSVQ) visualization. Based on MSVQ, the tile- and patch-level can be quantified simultaneously for slide-level pretraining and feature compression, respectively. The region data can be used to pretrain ABMIL via token index frequency matching.

Since the quantization operation in Equation 2 is non-differentiable, we adopt the straight-through gradient estimator [71], which copies gradients from the decoder input to the MLP encoder output for backpropagation.

The overall training objective of VQ is defined as:

$$\max \sum_{x \in M} \sum_{i=1}^{n} \cos(\mathbf{o}_i, \mathbf{h}_i) - \|\ell_2(\mathbf{e}_i) - \ell_2(\mathbf{v}_i)\|_2^2, \tag{3}$$

where M denotes the dataset of image tiles. For simplicity, we omit the straight-through gradient path and stop-gradient notation [71]. During optimization, the MLP encoder, codebook embeddings, and Transformer decoder are jointly trained to reconstruct the original spatial token representations.

2.2.2 Multi-Scale Vector Quantization

To simultaneously compress patch-level spatial token $\mathbf{ST}$ representations and generate an offline tokenizer for WSI self-supervised learning, we propose a **Multi-Scale Vector Quantization (MSVQ)** module. MSVQ encodes the FM tile $\mathbf{ST}$ features into K multi-scale discrete token maps $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K)$.

MSVQ builds upon the VQ architecture described in Section 2.2.1, with the key addition of a multi-scale quantization module. The encoding process is designed with residual paradigm [68, 38], as detailed in Algorithm 1.

Intuitively, when $\mathbf{R} = (\mathbf{r}_1)$ as scale ratio, MSVQ reduces to a standard VQ applied to the average-pooled patch token representation—akin to the tile-level [CLS] token. On the other hand, when $\mathbf{R} = (\mathbf{r}_K)$, MSVQ behaves identically to the patch-level VQ introduced in Section 2.2.1. The general form $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_K)$ enables vector quantization at multiple-scale semantic levels, including tile, patch, and intermediate resolutions. Please refer to Figure 2 for a visual illustration.

A shared codebook $\mathbf{Z}$ is employed across all scales, ensuring that tokens from each $\mathbf{r}_k$ are drawn from a unified vocabulary $[V_C]$. The decoding process mirrors the encoding pipeline in reverse order.

Algorithm 1: Multi-Scale VQ Encoding

Inputs: FM's spatial token feature $\mathbf{ST} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$ # $\mathbf{h}_i$ is the ViT outputs tokens despite of CLS;
Hyperparameters: number of scales K, resolutions $(H_k, W_k)_{k=1}^{K}$;
$\mathbf{f} \leftarrow \mathrm{Enc}(\mathbf{ST})$ is encoded feature, $\mathbf{R} \leftarrow []$ represents the residual list;
for $k = 1, \ldots, K$ **do**
  $\mathbf{r}_k \leftarrow \mathcal{Q}(\mathrm{interpolate}(\mathbf{f}, H_k, W_k))$, # get residual of resolution level k;
  $\mathbf{R} \leftarrow \mathrm{queue\_push}(\mathbf{R}, \mathbf{r}_k)$;
  $\mathbf{z}_k \leftarrow \mathrm{lookup}(\mathbf{Z}, \mathbf{r}_k)$;
  $\mathbf{z}_k \leftarrow \mathrm{interpolate}(\mathbf{z}_k, H_k, W_k)$;
  $\mathbf{f} \leftarrow \mathbf{f} - \mathbf{z}_k$;
Return: multi-scale tokens $\mathbf{R}$ and codebook indices $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_K]$;

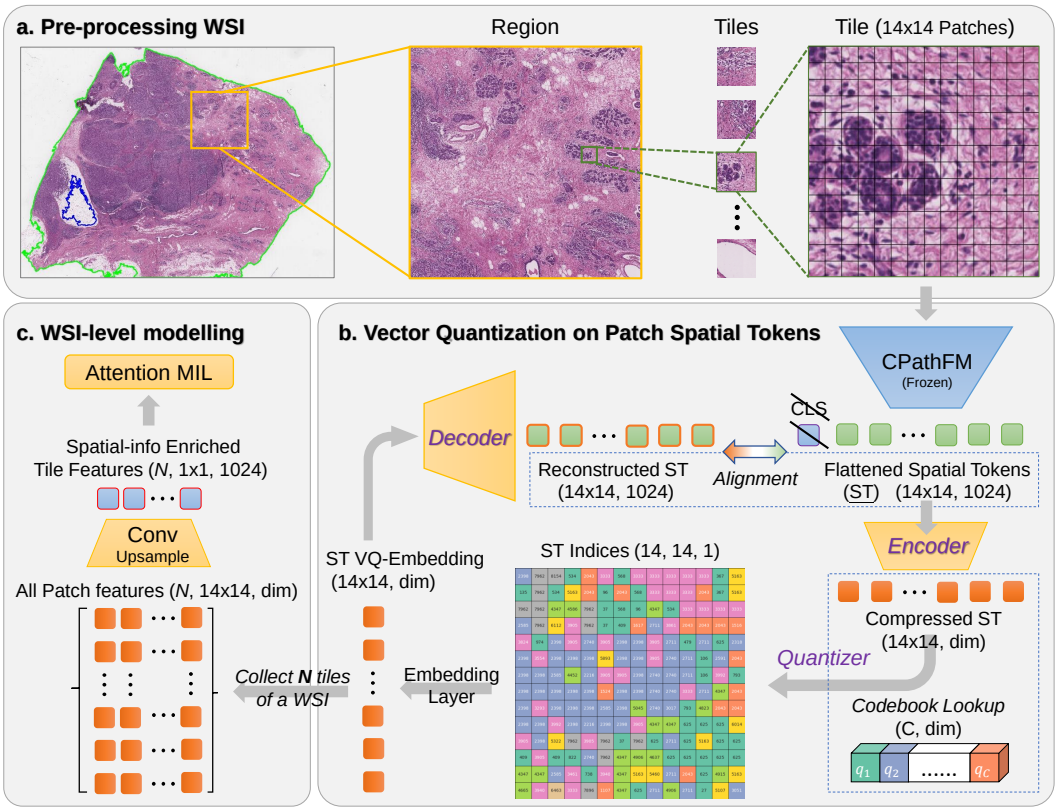

Figure 3: Overview of the proposed framework. (a) The pipeline for compressing spatial patch tokens using vector quantization (VQ) and multi-scale VQ (MSVQ). (b) Slide-level self-supervised learning (SSL) using MSVQ-generated tokenizers. (c) Downstream WSI task fine-tuning with compressed patch features.

2.3 Slide-Level Self-Supervised Learning

Leveraging the offline tokenizer generated by MSVQ for all WSI tiles, we design a self-supervised learning (SSL) pretraining framework tailored for WSI-MIL analysis. This framework is compatible with both mainstream MIL architectures, including attention-based MIL (ABMIL) and Transformer-based models.

2.3.1 ABMIL-Based Self-Supervised Learning

In supervised ABMIL training (e.g., WSI classification), adaptive pooling or max-pooling is typically employed to aggregate tile-level features for prediction at the WSI level. Inspired by this paradigm,

we formulate a simple yet effective SSL objective for ABMIL, grounded in the level-1 quantized indices from MSVQ (as shown in Figure 2).

Given a large-region crop from a WSI (e.g., a region of size 14336×14336, corresponding 4096 tiles each sized 224×224), the SSL objective for each region $\mathbf{x}$ is defined as:

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{c=1}^{C} q_c(\mathbf{x}) \log p_{\boldsymbol{\theta}}(\mathbf{x})_c, \tag{4}$$

where the soft target distribution $q_c(\mathbf{x})$ is computed based on the frequency of MSVQ token indices within the region, normalized over all C codebook categories. Specifically, $q_c(\mathbf{x})$ represents the proportion of tiles in region $\mathbf{x}$ assigned to token class c. The predicted probability $p_{\boldsymbol{\theta}}(\mathbf{x})_c$ for class c is obtained by passing the region through an ABMIL model followed by a classifier head with softmax activation:

$$p_{\boldsymbol{\theta}}(\mathbf{x})_c = \text{softmax}\left(\text{classifier}\left[\text{AttnPool}(\mathbf{x})\right]\right)_c. \tag{5}$$

2.3.2 WSI Transformer-Based Self-Supervised Learning

We adopt a masked image modeling (MIM) strategy inspired by MAE [25] and BEiT [53], but with a key difference: instead of raw image patches, we operate on pre-extracted feature representations as input. Given an input region composed of k tiles, represented as $\mathbf{x} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k\}$, we randomly mask a subset of tiles indexed by $\mathcal{M}$. The masked positions are replaced with a shared learnable embedding $\mathbf{e}_{[\text{M}]}$, and Rotary Positional Embedding (RoPE) [64] is applied to retain spatial coherence. The corrupted input becomes:

$$\mathbf{x}_{\text{corrupt}} = \{\mathbf{t}_1, \cancel{\mathbf{t}_2}, \ldots, \cancel{\mathbf{t}_i}, \mathbf{t}_{i+1}, \ldots, \mathbf{t}_k\}. \tag{6}$$

For each masked tile, a softmax classifier is trained to predict the corresponding token index, which is obtained from the level-1 quantized output of the MSVQ tokenizer (see Section 2.2.2). This provides a discrete and consistent supervision signal.

The training objective is formulated as:

$$\mathcal{L}_{\text{mask-modeling}}(\boldsymbol{\theta}) = -\sum_{\mathbf{x} \in \mathcal{D}} \sum_{i \in \mathcal{M}} \log p_{\boldsymbol{\theta}}(z_i \mid \mathbf{x}_i^{\mathcal{M}}), \tag{7}$$

where z_i denotes the MSVQ token index for the i-th masked tile, and $\mathcal{D}$ is the dataset of training regions. Compared to the online tokenizer used in iBOT and related frameworks, our MSVQ-based offline tokenizer provides a more stable and reliable supervisory signal for SSL pretraining.

2.4 WSI Downstream-Task Fine-Tuning

As illustrated in Figure 3c, the refined WSI input consists of patch feature embeddings with a compressed shape of $(N, 14, 14, \text{dim})$, where N represents the total number of tiles in a WSI, and $(14, 14)$ corresponds to the standard 2D patch arrangement in ViT for each tile. To enhance the feature representation for downstream tasks, we first apply convolutional layers (Convs) with upsampling, increasing the output channel size while reducing to fewer tokens to better capture task-relevant information. The extracted features are then reshaped to match the original CLS token representation of tile-based ViT.

It is notable that the encoding direction of $\mathbf{ST}$ is not so controllable since the the encoded embeddings are in the middle layers (after encoding, before decoding). To keep its feature space as original, we align the output of Convs to original level-1 tile feature during VQ pre-training. This module will be further fine-tuned during slide-level task. Finally, the processed features are fed into downstream MIL models, including both ABMIL and WSI-Transformer (see Section 2.1).

2.5 Overall Framework and Implementation

We summarize the overall framework of our method below. The WSI pre-processing follows the approach used in previous work [49]:

- **Patch-Level VQ Learning (Figure 3b)**: This module aims to compress all patch token features from FMs, making them trainable for downstream tasks. Multi-scale VQ learning (Figure 2) further enables slide-level SSL supervision and subsumes patch-level VQ learning.
- **Slide-Level SSL (Figure 2)**: Leveraging the tile-level tokenizer learned via MSVQ, SSL can be effectively applied to both ABMIL and WSI-Transformer models.
- **WSI Downstream-Task Fine-Tuning (Figure 3c)**: Fine-tuning serves two purposes: (a) transforming patch features into a more suitable representation for downstream tasks, and (b) fine-tuning the pretrained slide-level SSL model for improved performance.

3 Experiments

In this section, we evaluate the performance of the proposed method and compare it with various baselines. Additionally, we conduct ablation studies to further analyze its effectiveness.

3.1 Pretraining Implementation Details

VQ Pretraining: We conduct VQ pretraining on 1M randomly cropped 224×224 tiles extracted from all TCGA [69] diagnostic pathology WSIs. During training, the FM backbone (e.g., UNI with ViT-Large) remains frozen. The codebook has a size of $C = 8192$ with an embedding dimension of 16. For MSVQ, we employ a multi-scale resolution list: $\{1 \times 1, 2 \times 2, 4 \times 4, 7 \times 7, 14 \times 14\}$. The VQ encoder, decoder, and codebook are frozen after pretraining. The model is trained on 4 RTX-3090 GPUs for 50 epochs using a batch size of 128 tile images per GPU. The total training time is approximately 22 hours.

WSI-SSL Pretraining: We crop all TCGA diagnostic WSIs into regions of resolution 3584×3584, yielding a dataset of approximately 250k regions. To facilitate SSL, a pretrained MSVQ model is used to extract the quantized indices of each tile within a region, requiring only about 65MB for storage.

During pretraining, the indices of each region are first re-embedded via a frozen VQ module, resulting in a feature representation of shape $(256, 14 \times 14, 16)$. The convolutional module consists of four conv layers with a stride of 2, progressively increasing the output channels from $128 \rightarrow 256 \rightarrow 512 \rightarrow 1024$. This process transforms the features into spatially enriched embeddings of shape $(256, 1 \times 1, 1024)$. These embeddings are subsequently fed into either an ABMIL model or a 6-layer WSI-Transformer for pretraining.

3.2 Downstream Tasks

We primarily focus on WSI classification and survival prediction. For dataset details, please refer to Appendix A.2. For illustration purpose, we also run two experiments on ROI classification to clarify [CLS] token is not all we need.

The data processing and embedding procedure are identical to the region-based approach but are applied at the WSI level. The (x, y) coordinates of tiles are also stored to facilitate positional encoding in the WSI-Transformer.

During fine-tuning, both the convolutional module and the WSI model are trained with a batch size of 1 for 20 epochs. The learning rate is fixed at 1×10^{-4}, with a weight decay of 1×10^{-4}, using the AdamW optimizer with default settings.

For ABMIL, both randomly initialized and pretrained models are fine-tuned using the same hyper-parameters and training protocol. For WSI-Transformer, LoRA [27] adaptation (ap-

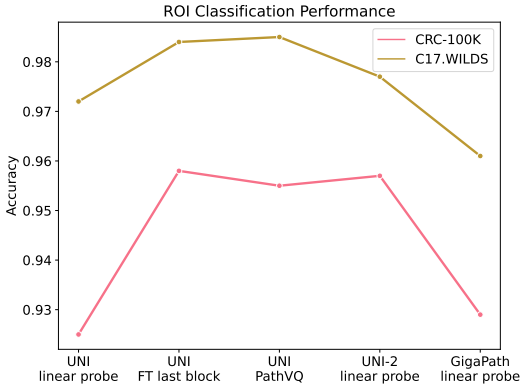

Figure 4: ROI classification. Obviously, further fine-tuning (FT) on the last block of UNI ViT can further improve the downstream results. PathVQ, by compressing and reconstructing the patch spatial token, can achieve comparable improvement. UNI-2, however, does not show consistency improvement compared to FT and PathVQ.

plied to all `nn.Linear` layers) is used with $rank = 16$ during fine-tuning of the pretrained model to mitigate overfitting. For Transformer initialized from scratch, full fine-tuning is employed.

3.2.1 Tile/ROI Classification

We evaluate tile/ROI classification performance using dataset CRC-100K [35] (9 categories) and Camelyon-17 WILDS [36] tiles (binary). The result in shown in Figure 4: By only updating the last Transformer block of UNI, the result can be significantly improved. Our PathVQ method is also included and shown comparable improvement to FT. All these results are better than linear probe (freeze backbone and fed [CLS] token feature to classification head). UNI-2, also using linear-probe seems can not scaling up with strong performance on every down-stream tasks.

3.2.2 WSI Tumor Classification

Table 1: Slide-Level Tumor Classification based on FM. The results in the first-row are all trained on UNI, while the second-row we include some recent stronger FMs. The cyan rows are our methods including PathVQ and Slide-level Pre-Training (SPT). The orange rows demonstrate how much (Δ) of our PathVQ method and UNI-2 improved over UNI with ABMIL setting. The **bold** and underline denote the best and second-best result, respectively.

	Tumor classification		Mutation Prediciton	
	BRACS		LGG-GBM	
Method	F1	AUC	F1	AUC
CLAM-SB [49]	0.640±0.05	0.844±0.03	0.672±0.06	0.842±0.03
DTFD-MIL [84]	0.655±0.03	0.878±0.02	0.697±0.04	0.857±0.02
TransMIL [61]	0.592±0.03	0.859±0.02	0.678±0.05	0.847±0.03
ABMIL [31]	0.692±0.03	0.875±0.02	0.685±0.07	0.852±0.04
+PathVQ	0.730±0.02	0.902±0.01	0.723±0.04	0.871±0.04
Δ over UNI + ABMIL	3.8% ↑	2.7% ↑	4.8% ↑	1.9% ↑
+PathVQ + SPT	0.747±0.01	0.906±0.01	0.752±0.03	0.879±0.02
Roformer	0.678±0.03	0.882±0.01	0.675±0.03	0.861±0.02
+PathVQ	0.711±0.02	0.892±0.01	0.739±0.04	0.872±0.02
+PathVQ + SPT	**0.754±0.02**	**0.910±0.01**	**0.758±0.02**	**0.886±0.01**
UNI-2 + ABMIL	0.698±0.03	0.887±0.02	0.699±0.03	0.859±0.01
Δ over UNI + ABMIL	0.6% ↑	1.2% ↑	1.4% ↑	0.7% ↑
GigaPath	0.677±0.03	0.862±0.03	0.703±0.04	0.864±0.02
TITAN	0.696±0.04	0.891±0.01	0.711±0.03	0.868±0.02

We first evaluate our method on the BRACS [4], a dataset with three categories—negative, benign, and malignant cancer. We then evaluate on TCGA LGG-GBM [69] focus on R132 [2] gene mutation as binary classification. (We notice that popular-used WSI binary tumor classification tasks(e.g. Camelyon [47], TCGA-NSCLC [69]) are nearly solved (AUC>97) given FMs progress. So here we mainly focus on more difficult task, like more categories, and will explore and validate more difficult datasets in near future.)

Compared Baselines: Since our method primarily focuses on extracting improved tile-level features for WSI analysis, we compare it against various WSI analysis models with different architectural designs: ABMIL [31], DSMIL [39] (introduces a max-pooling branch alongside the attention mechanism), and DTFD-MIL [84] (employs sub-bags for hierarchical learning). TransMIL [61] (leveraging Nyström self-attention [78] for computational efficiency), Transformer with 2-d RoPE [64, 41, 54].

FMs like GigaPath (a 12-layers WSI-Transformer (efficiently implemented using LongViT [74]), pretrained on large-scale private data via MAE [25], with a [CLS] token as tile feature), TITAN [79, 17] (a 6-layers WSI-Transformer with 2D-ALiBi positional encoding [55, 41], pretrained on large-scale private data using iBOT [88], with Conch-v1.5 as the tile feature extractor ([CLS] token).), and UNI-2 are also included. Certain works that focus on orthogonal aspects, such as overfitting

mitigation, hard instance mining, etc. [89, 86, 87, 56, 66, 67, 14, 45, 80], are not included in our primary comparison.

For all the experiments, we report the macro-AUC and macro-F1 scores (over five-runs or five-fold cross validation) because of class imbalance.

WSI Classification Results Analysis: The results are reported in Table 1. We can first observe that ABMIL and Roformer show significant improvement when combined with our PathVQ compressor into UNI. The results difference of UNI+PathVQ+ABMIL (about 3% improvement with adding 1M tile data) and UNI2+ABMIL (about 1% improvement with adding large-scale (>>1M) of tile data, and $2 \sim 6\times$ model size) demonstrate that the scalability of previous FMs are bottlenecked by the [CLS] token information losses. In addition, the results of our slide-level pretraining (**SPT**) also show consistency improvement compared with random initialization.

3.2.3 WSI Survival Prediction

Table 2: **Survival prediction** Results of PathVQ and baselines for measuring patient disease-specific survival. All methods in Prototype and MIL use UNI features [10]. Best performance in **bold**, second best underlined.

	TCGA	BRCA	CRC	BLCA	UCEC	KIRC
Prototype (unsup. cox loss)	H2T [73]	0.672±0.07	0.639±0.11	0.566±0.05	0.715±0.09	0.703±0.11
	OT [50]	0.755±0.06	0.622±0.09	0.603±0.04	0.747±0.08	0.695±0.09
	PANTHER [63]	0.758±0.06	0.665±0.10	0.612±0.07	0.757±0.10	0.716±0.10
MIL (supervised. UNI)	AttnMISL [81]	0.627±0.08	0.639±0.10	0.485±0.06	0.581±0.12	0.649±0.09
	ILRA [77]	0.649±0.10	0.555±0.10	0.550±0.04	0.632±0.02	0.637±0.14
	TransMIL [61]	0.612±0.07	**0.684±0.06**	0.595±0.06	0.695±0.08	0.671±0.10
	ABMIL [31] †	0.644±0.05	0.608±0.09	0.550±0.06	0.669±0.07	0.684±0.06
	ABMIL reproduce	0.633±0.06	0.612±0.08	0.540±0.07	0.671±0.08	0.691±0.08
	+ PathVQ	0.655±0.05	0.649±0.12	0.608±0.05	0.721±0.10	0.760±0.08
	Δ over ABMIL	2.2% ↑	3.7% ↑	6.8% ↑	5.0% ↑	6.9% ↑
	+PathVQ + SPT	**0.674±0.06**	0.659±0.08	**0.616±0.05**	**0.748±0.11**	**0.778±0.08**
	Roformer	0.602±0.09	0.617±0.13	0.572±0.07	0.721±0.08	0.655±0.13
	+PathVQ	0.644±0.07	0.587±0.09	0.597±0.05	0.741±0.09	0.748±0.09
	+PathVQ + SPT	0.673±0.07	0.679±0.08	0.603±0.05	0.734±0.11	0.765±0.08
UNI-2	ABMIL	0.614±0.02	0.618±0.11	0.539±0.08	0.672±0.08	0.659±0.11
Slide-FMs (SOTA, ckpt-only)	CHIEF [76]	0.737±0.04	0.680±0.08	0.599±0.02	0.758±0.10	0.736±0.06
	GigaPath [79]	0.687±0.08	0.628±0.08	0.589±0.05	0.779±0.10	0.751±0.07
	TITAN [16] (cox loss)	0.713±0.04	0.710±0.11	0.657±0.05	0.789±0.09	0.774±0.06

We evaluate survival prediction on five TCGA datasets: BRCA, BLCA, CRC, UCEC, and KIRC. The model is trained using the negative log-likelihood (NLL, notice that some compared models' are trained via Cox-loss generally gain better result, please check Appendix A.6 for details.) loss and evaluated using the c-index with 5-fold cross validation (the result of last epoch is reported).

For fair comparison, we follow the default training pipeline of PANTHER [63], including hyper-parameters and data splits, and integrate our proposed model modifications along with pretrained weights.

Compared Baselines: We categorize the baselines into three groups: **Unsupervised Prototype-Based Approaches:** H2T [73] (clusters tile embeddings and pools them within each cluster), OT [50] (aggregates patch features into a set of prototypes using Optimal Transport), PANTHER [63] (models prototype tile embeddings via a Gaussian Mixture Model). **Supervised MIL Models:** AttnMISL [81] (combines prototype-based learning with MIL), ABMIL [31], TransMIL [61], ILRA [77], and Transformer with RoPE [64]. **Slide-Level FMs:** CHIEF [76]: A large-scale ABMIL-pretrained model using contrastive learning to predict organ source, with CTransPath as the tile feature extractor (mean-pooled features), GigaPath [79] and TITAN [17]. UNI-2 [10] feature extractor with ABMIL model.

Survival Prediction Results Analysis: The results are reported in Table 2. We can observe that ABMIL and Roformer show significant improvement when combined with our PathVQ compressor into UNI. But for Roformer with large-scale of parameters, the performance get easily overfitting to

labels thus showing interior performance, which demonstrate the necessity of pretraining. For UNI-2, the improvement is marginal, proving our previous claim on the [CLS] bottleneck of scalability.

3.3 Ablations and Visualization

Performance vs. Complexity across reduction methods is shown in Figure 5. PCA: Reduces token dimensions by projecting them onto a low-rank linear subspace, preserving variance but potentially discarding discriminative details.The F1-score reaches about 0.70.

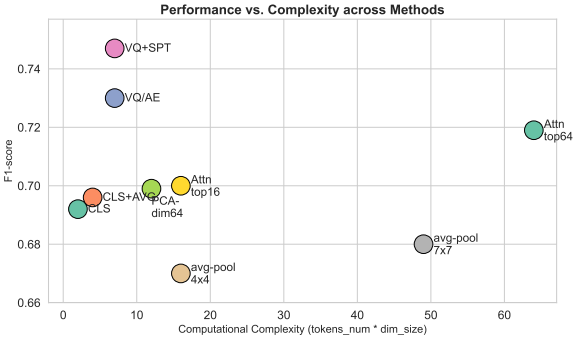

Figure 5: Performance–complexity trade-off across various token reduction strategies on BRACS. Our proposed VQ method achieves the best balance between accuracy and efficiency, delivering the highest F1-score (0.73) with significantly reduced token complexity. Other approaches, including PCA, average pooling, and CLS-based methods, show varying trade-offs.

Average Pooling: Aggregates tokens by computing their mean, leading to compact representations but often oversmoothing critical local information. We have conducted pooling original 196=14x14 token features into 4x4 and 7x7 tokens, resulting in about 10 and 2 times token reduction. However, the result only show similar performance compared to original CLS token.

CLS and overall average-pooling: as performed in virchow, the CLS and average-pooling token can complement a little to each other and resulting in around 1.0 point improvement.

We also performed token selection via CLS token attention top-k ranking. The selected top64 can reach a F1-score around 0.72 but is still too heavy in computation cost. The selected top16 can reach a F1-score of 0.70, can be seen as a trade-off, but lose too much on performance.

Our proposed VQ-based method achieves the best balance, attaining the highest F1-score of 0.73 while significantly reducing token complexity. In contrast, other strategies demonstrate varying degrees of compromise between accuracy and computational cost.

Convs pretraining ablation: Please check Appendix Figure 6.

For additional ablation studies and visualizations, please refer to the Appendix A.4. The main findings: The VQ reconstruction performance remains relatively stable when varying the quantized embedding dimension (32) and codebook size (16384). The reconstruction results of MSVQ show improvements.

4 Conclusion and Limitations

In this work, we introduced a novel vector quantization (VQ) distillation framework to address the inherent bottleneck of existing computational pathology foundation models in whole-slide image analysis. Furthermore, our multi-scale VQ strategy unifies patch- and tile-level features, not only improving feature reconstruction but also serving as an effective self-supervised learning supervision target for slide-level pretraining. The main **limitation** is that the VQ learning process need extra training data. And the VQ still lose some information though it is acceptable. By efficiently compressing patch-level spatial tokens while preserving critical spatial and contextual information, our method significantly reduces storage and computational costs without compromising performance.

5 Acknowledgments

This study was partially supported by "Pioneer" and "Leading Goose" R&D Program of Zhejiang (Grant 2025SDXHDX0003), the National Natural Science Foundation of China (Grant No.62506306), and foundation of Muyuan Laboratory (Program ID: 14106022401,14106022402).

References

[1] Benjamin Bergner, Christoph Lippert, and Aravindh Mahendran. Iterative patch selection for high-resolution image recognition. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] Fonnet E Bleeker, Nadia A Atai, Simona Lamba, Ard Jonker, Denise Rijkeboer, Klazien S Bosch, Wikky Tigchelaar, Dirk Troost, W Peter Vandertop, Alberto Bardelli, et al. The prognostic idh1 r132 mutation is associated with reduced nadp+-dependent idh activity in glioblastoma. *Acta neuropathologica*, 119:487–494, 2010.

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[4] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, Maria Gabrani, Florinda Feroce, and Maria Frucci. Bracs: A dataset for breast carcinoma subtyping in h&e histology images, 2021.

[5] Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021.

[7] Tsai Hor Chan, Fernando Julio Cendra, Lan Ma, Guosheng Yin, and Lequan Yu. Histopathology whole slide image analysis with heterogeneous graph representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15661–15670, 2023.

[8] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[9] Richard J. Chen and et al. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *CVPR*, pages 16144–16155, June 2022.

[10] 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.

[11] Richard J. Chen, Ming Y. Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y. Chen, Drew F. K. Williamson, and Faisal Mahmood. Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks, 2021.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[13] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[14] Yufei CUI, Ziquan Liu, Xiangyu Liu, Xue Liu, Cong Wang, Tei-Wei Kuo, Chun Jason Xue, and Antoni B. Chan. Bayes-MIL: A new probabilistic perspective on attention-based multiple instance learning for whole slide images. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] 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.

[16] Tong Ding, Sophia J Wagner, Andrew H Song, Richard J Chen, Ming Y Lu, Andrew Zhang, Anurag J Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, et al. Multimodal whole slide foundation model for pathology. *arXiv preprint arXiv:2411.19666*, 2024.

[17] Tong Ding, Sophia J. Wagner, Andrew H. Song, Richard J. Chen, Ming Y. Lu, Andrew Zhang, Anurag J. Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, Drew F. K. Williamson, Bowen Chen, Cristina Almagro-Perez, Paul Doucet, Sharifa Sahai, Chengkuan Chen, Daisuke Komura, Akihiro Kawabe, Shumpei Ishikawa, Georg Gerber, Tingying Peng, Long Phi Le, and Faisal Mahmood. Multimodal whole slide foundation model for pathology, 2024.

[18] William H Equitz. A new vector quantization clustering algorithm. *IEEE transactions on acoustics, speech, and signal processing*, 37(10):1568–1575, 1989.

[19] Christopher Fifty, Ronald Guenther Junkins, Dennis Duan, Aniketh Iyengar, Jerry Weihong Liu, Ehsan Amid, Sebastian Thrun, and Christopher Re. Restructuring vector quantization with the rotation trick. In *The Thirteenth International Conference on Learning Representations*, 2025.

[20] 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*, 2023.

[21] Olga Fourkioti, Matt De Vries, and Chris Bakal. CAMIL: Context-aware multiple instance learning for cancer detection and subtyping in whole slide images. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29, 1984.

[23] Yonghang Guan, Jun Zhang, Kuan Tian, Sen Yang, Pei Dong, Jinxi Xiang, Wei Yang, Junzhou Huang, Yuyao Zhang, and Xiao Han. Node-aligned graph convolutional network for whole-slide image representation and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18813–18823, 2022.

[24] Metin N Gurcan, Laura E Boucheron, Ali Can, Anant Madabhushi, Nasir M Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE reviews in biomedical engineering*, 2:147–171, 2009.

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021.

[26] Xinhai Hou, Cheng Jiang, Akhil Kondepudi, Yiwei Lyu, Asadur Chowdury, Honglak Lee, and Todd C Hollon. A self-supervised framework for learning whole slide representations. *arXiv preprint arXiv:2402.06188*, 2024.

[27] 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.

[28] Shigao Huang, Jie Yang, Simon Fong, and Qi Zhao. Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges. *Cancer letters*, 471:61–71, 2020.

[29] 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.

[30] Wisdom Ikezogwo, Saygin Seyfioglu, Fatemeh Ghezloo, Dylan Geva, Fatwir Sheikh Mohammed, Pavan Kumar Anand, Ranjay Krishna, and Linda Shapiro. Quilt-1m: One million image-text pairs for histopathology. *Advances in neural information processing systems*, 36:37995–38017, 2023.

[31] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2127–2136. PMLR, 10–15 Jul 2018.

[32] Guillaume Jaume, Paul Doucet, Andrew Song, Ming Yang Lu, Cristina Almagro Pérez, Sophia Wagner, Anurag Vaidya, Richard Chen, Drew Williamson, Ahrong Kim, et al. Hest-1k: A dataset for spatial transcriptomics and histology image analysis. *Advances in Neural Information Processing Systems*, 37:53798–53833, 2025.

[33] Guillaume Jaume, Lukas Oldenburg, Anurag Vaidya, Richard J Chen, Drew FK Williamson, Thomas Peeters, Andrew H Song, and Faisal Mahmood. Transcriptomics-guided slide representation learning in computational pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9632–9644, 2024.

[34] Syed Ashar Javed, Dinkar Juyal, Harshith Padigela, Amaro Taylor-Weiner, Limin Yu, and aaditya prakash. Additive MIL: Intrinsically interpretable multiple instance learning for pathology. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[35] Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue, April 2018.

[36] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

[37] Tristan Lazard, Marvin Lerousseau, Etienne Decencière, and Thomas Walter. Giga-ssl: Self-supervised learning for gigapixel images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4305–4314, 2023.

[38] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[39] Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2021.

[40] Hao Li, Ying Chen, Yifei Chen, Rongshan Yu, Wenxian Yang, Liansheng Wang, Bowen Ding, and Yuchen Han. Generalizable whole slide image classification with fine-grained visual-semantic interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11398–11407, 2024.

[41] Honglin Li, Yunlong Zhang, Pingyi Chen, Zhongyi Shui, Chenglu Zhu, and Lin Yang. Rethinking transformer for long contextual histopathology whole slide image analysis. *arXiv preprint arXiv:2410.14195*, 2024.

[42] Honglin Li, Chenglu Zhu, Yunlong Zhang, Yuxuan Sun, Zhongyi Shui, Wenwei Kuang, Sunyi Zheng, and Lin Yang. Task-specific fine-tuning via variational information bottleneck for weakly-supervised pathology whole slide image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7454–7463, June 2023.

[43] Jiawen Li, Yuxuan Chen, Hongbo Chu, Qiehe Sun, Tian Guan, Anjia Han, and Yonghong He. Dynamic graph representation with knowledge-aware attention for histopathology whole slide image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11323–11332, 2024.

[44] Ruoyu Li, Jiawen Yao, Xinliang Zhu, Yeqing Li, and Junzhou Huang. Graph cnn for survival analysis on whole slide pathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 174–182. Springer, 2018.

[45] Tiancheng Lin, Zhimiao Yu, Hongyu Hu, Yi Xu, and Chang-Wen Chen. Interventional bag multi-instance learning on whole-slide pathological images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19830–19839, 2023.

[46] Lucas D Lingle. Transformer-vq: Linear-time transformers via vector quantization. *arXiv preprint arXiv:2309.16354*, 2023.

[47] Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, Oscar Geessink, Maschenka Balkenhol, Peter Bult, Altuna Halilovic, Meyke Hermsen, Rob van de Loo, Rob Vogels, et al. 1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. *GigaScience*, 7(6):giy065, 2018.

[48] 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.

[49] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering*, 5(6):555–570, 2021.

[50] Grégoire Mialon, Dexiong Chen, Alexandre d'Aspremont, and Julien Mairal. A trainable optimal transport embedding for feature aggregation and its relationship to attention. *arXiv preprint arXiv:2006.12065*, 2020.

[51] OpenAI. Gpt-4 technical report. *arXiv*, 2023.

[52] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[53] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

[54] Etienne Pochet, Rami Maroun, and Roger Trullo. Roformer for position aware multiple instance learning in whole slide image classification, 2023.

[55] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[56] Linhao Qu, xiaoyuan Luo, Manning Wang, and Zhijian Song. Bi-directional weakly supervised knowledge distillation for whole slide image classification. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[58] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[59] Charlie Saillard, Rodolphe Jenatton, Felipe Llinares-López, Zelda Mariet, David Cahané, Eric Durand, and Jean-Philippe Vert. H-optimus-0, 2024.

[60] Wei Shao, Tongxin Wang, Zhi Huang, Zhi Han, Jie Zhang, and Kun Huang. Weakly supervised deep ordinal cox model for survival prediction from whole-slide pathological images. *IEEE Transactions on Medical Imaging*, 40(12):3739–3747, 2021.

[61] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, and yongbing zhang. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2136–2147. Curran Associates, Inc., 2021.

[62] Jiangbo Shi, Chen Li, Tieliang Gong, Yefeng Zheng, and Huazhu Fu. Vila-mil: Dual-scale vision-language multiple instance learning for whole slide image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11248–11258, 2024.

[63] Andrew H Song, Richard J Chen, Tong Ding, Drew FK Williamson, Guillaume Jaume, and Faisal Mahmood. Morphological prototyping for unsupervised slide representation learning in computational pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11566–11578, 2024.

[64] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2021.

[65] Yuxuan Sun, Yunlong Zhang, Yixuan Si, Chenglu Zhu, Kai Zhang, Zhongyi Shui, Jingxiong Li, Xuan Gong, XINHENG LYU, Tao Lin, et al. Pathgen-1.6 m: 1.6 million pathology image-text pairs generation through multi-agent collaboration. In *The Thirteenth International Conference on Learning Representations*.

[66] Wenhao Tang, Sheng Huang, Xiaoxian Zhang, Fengtao Zhou, Yi Zhang, and Bo Liu. Multiple instance learning framework with masked hard instance mining for whole slide image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4078–4087, 2023.

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

[68] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2025.

[69] 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.

[70] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[71] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[72] Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Siqi Liu, Kristen Severson, Eric Zimmermann, James Hall, Neil Tenenholtz, et al. Virchow: A million-slide digital pathology foundation model. *arXiv preprint arXiv:2309.07778*, 2023.

[73] Quoc Dang Vu, Kashif Rajpoot, Shan E Ahmed Raza, and Nasir Rajpoot. Handcrafted histological transformer (h2t): Unsupervised representation of whole slide images. *Medical image analysis*, 85:102743, 2023.

[74] Wenhui Wang, Shuming Ma, Hanwen Xu, Naoto Usuyama, Jiayu Ding, Hoifung Poon, and Furu Wei. When an image is worth 1,024 x 1,024 words: A case study in computational pathology. *arXiv preprint arXiv:2312.03558*, 2023.

[75] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Junzhou Huang, Wei Yang, and Xiao Han. Transpath: Transformer-based self-supervised learning for histopathological image classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 186–195. Springer, 2021.

[76] Xiyue Wang, Junhan Zhao, Eliana Marostica, Wei Yuan, Jietian Jin, Jiayu Zhang, Ruijiang Li, Hongping Tang, Kanran Wang, Yu Li, et al. A pathology foundation model for cancer diagnosis and prognosis prediction. *Nature*, 634(8035):970–978, 2024.

[77] Jinxi Xiang and Jun Zhang. Exploring low-rank property in multiple instance learning for whole slide image classification. In *The Eleventh International Conference on Learning Representations*, 2023.

[78] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[79] Hanwen Xu, Naoto Usuyama, Jaspreet Bagga, Sheng Zhang, Rajesh Rao, Tristan Naumann, Cliff Wong, Zelalem Gero, Javier González, Yu Gu, et al. A whole-slide foundation model for digital pathology from real-world data. *Nature*, pages 1–8, 2024.

[80] Shu Yang, Yihui Wang, and Hao Chen. Mambamil: Enhancing long sequence modeling with sequence reordering in computational pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 296–306. Springer, 2024.

[81] Jiawen Yao, Xinliang Zhu, Jitendra Jonnagaddala, Nicholas Hawkins, and Junzhou Huang. Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks. *Medical Image Analysis*, 65:101789, 2020.

[82] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[83] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[84] Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E. Coupland, and Yalin Zheng. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. *ArXiv*, abs/2203.12081, 2022.

[85] Jingwei Zhang, Saarthak Kapse, Ke Ma, Prateek Prasanna, Joel Saltz, Maria Vakalopoulou, and Dimitris Samaras. Prompt-mil: Boosting multi-instance learning schemes via task-specific prompt tuning, 2023.

[86] Yunlong Zhang, Honglin Li, Yuxuan Sun, Sunyi Zheng, Chenglu Zhu, and Lin Yang. Attention-challenging multiple instance learning for whole slide image classification. *arXiv preprint arXiv:2311.07125*, 2023.

[87] Yunlong Zhang, Zhongyi Shui, Yunxuan Sun, Honglin Li, Jingxiong Li, Chenglu Zhu, Sunyi Zheng, and Lin Yang. Adr: Attention diversification regularization for mitigating overfitting in multiple instance learning based whole slide image classification. *arXiv preprint arXiv:2406.15303*, 2024.

[88] 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.

[89] Wenhui Zhu, Xiwen Chen, Peijie Qiu, Aristeidis Sotiras, Abolfazl Razi, and Yalin Wang. Dgr-mil: Exploring diverse global representation in multiple instance learning for whole slide image classification. In *European Conference on Computer Vision*, pages 333–351. Springer, 2024.

[90] Eric Zimmermann, Eugene Vorontsov, Julian Viret, Adam Casson, Michal Zelechowski, George Shaikovski, Neil Tenenholtz, James Hall, David Klimstra, Razik Yousfi, et al. Virchow2: Scaling self-supervised mixed magnification models in pathology. *arXiv preprint arXiv:2408.00738*, 2024.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction are accurately reflect the paper's contributions and scope, matching the empirical results.

   Guidelines:

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

2. **Limitations**

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

   Answer:[Yes]

   Justification: The main weakness is discussed in last section: need further pre-training, the training data mainly include TCGA.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We provide patch and tile visualizations to support the theoretical multi-scale Vector Quantization learning.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide comprehensive details of dataset selection and pre-processing, together with training details. We will release full code implementation.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: All our data is public and use 'CLAM' for pre-processing. We will release full code implementation.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We provide model backbone selection and implementation details in core paper and further provide data split and pre-processing in supplemental material.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: For each experiments we have reported the mean and std via multi-runs or cross-folds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Our memory efficiency method is run on RTX3090 GPU (24g), as detailed in core paper and appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and understood the policies, and we believe that neither the manuscript nor the study violates any of these.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: There is no negative societal impact of our paper. Our work focus on computer aided diagnosis for potential medical use, which is discussed in conclusions of core paper.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our work mainly focus on recognition for real-world medical pathology image, without any generative problem.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All the data and tools are public accessible and well referenced.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: There is no new assets right now.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: No human subjects were directly involved in this study. The research was conducted using publicly available datasets that have been previously collected with appropriate ethical approvals and anonymization. Therefore, no additional IRB approval was required.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Supplementary Materials

A.1 Related Work

A.1.1 Pathological Whole Slide Image Analysis

Whole Slide Images (WSIs) contain a wealth of visual information that plays a crucial role in pathological analysis [5, 49]. However, obtaining detailed cell-level annotations is both labor-intensive and time-consuming [5, 49, 9], posing a significant challenge for large-scale WSI analysis. To address this, weakly-supervised learning has emerged as a promising direction in computational pathology.

Computational Pathology Foundation Models The performance of early WSI MIL approaches [31, 11, 39, 49, 84, 34, 56, 14, 1, 9, 61, 41, 11, 44, 23, 7, 21] relied heavily on tile-level features extracted from pre-trained models [42, 20]. To address this, FMs have been developed and shown significant advancements in both tile-level [75, 48, 10, 20, 72, 59] and WSI-level analysis [79, 17]. These FMs leverage visual Self-supervised Learning (SSL) techniques [6, 52, 13, 12] on large-scale unlabeled datasets [75, 10, 20] or organize pathology image-text pairs to learn multimodal representation [57, 17, 48]. FMs have demonstrated superior performance in downstream tasks such as cancer subtyping, survival prediction, and biomarker identification.

Recently, authors in Hest-1k [32] observe that tile encoders like Conch [48] can be further fine-tuned to obtain better downstream tile task result. However, the key challenge that the computational cost high-resolution WSIs makes fine-tuning [85, 42] FMs with overwhelm parameters difficult. Most approaches [48, 10, 20] resort to using pretrained [CLS] token representation of tile-level FM as slide-level inputs, which may lead to the loss of critical spatial information. Some models, such as UNI-2 [10], attempt to scale up ViTs into larger-size as tile encoders to extract better feature representations, but only achieve marginal improvements from ViT-Large to ViT-Giant. We argue that this performance bottleneck stems from the spatial information loss inherent in [CLS] token representations. Other efforts, such as Virchow-2 [90], find that combining [CLS] tokens with [AVG] (average pooling of all spatial tokens) can yield some improvement (less than 1 point). Motivated by these findings, we propose to keep but compress all spatial patch tokens and further extract useful information for downstream WSI tasks analysis.

Slide-level FMs / SSL pretraining: Some recent slide-level FMs, e.g. GigaPath [79] and TITAN [17] are modeled via Transformer with 6 to 12 layers, then pretrained via MAE [25] and iBOT [88] respectively. Some other slide SSL models [26, 37, 9] also propose to employ slide-level augmentation with contrastive learning (CL) [12, 52] for pretraining. But there are some training problems of these works: 1) The main augmentations in slide-level to generate different views for CL is limited, like crop or random-drop, since the tile feature are pre-processed and stored. This hinders the performance of CL pretraining. 2) The self-supervised target. The iBOT, [88] used in TITAN [17] predicting masked token to match online-tokenizer, is not so stable during training. The MAE in GigaPath [79] need to regress the feature of masked tiles, which may be too difficult to fit and hinder downstream tasks. The CHIEF [76], on the other way, pretrain ABMIL by constrastively predicting tumor organ source (extra information).

Different to these work, in this paper we train a offline tile tokenizer via vector quantization which can offer self-supervision for both WSI-Transformer mask modeling and ABMIL. This is more stable for pretraining and need no further information.

A.1.2 Vector Quantization

Vector quantization (VQ) [22] is a fundamental technique in signal processing and machine learning, widely used for data compression, clustering, and generative modeling [18, 71, 58]. Recent developments in deep learning have led to neural vector quantization methods, such as Vector Quantized Variational Autoencoders (VQ-VAEs) [71, 58, 8] and quantized transformers [46], which integrate VQ into end-to-end learning pipelines to enhance expressiveness and efficiency. To further improve VQ, techniques such as residual quantization [38] and rotation tricks [19] have been proposed. Recent studies [82, 83] reveal that lower-dimensional quantized vectors (dimension size ranging from 8 to 32) can improve codebook usage and reconstruction performance, providing strong compression capabilities that benefit this study. Unlike recent VQ methods that focus primarily on visual genera-

tion [71, 68, 58], our work focus on feature compression and distillation of pretrained pathology tile encoder features via a VQ quantizer.

A.2 Data Description

BReAst Carcinoma Subtyping (BRACS) [4] collect H&E stained Histology Images, containing 547 WSIs for three lesion types, i.e., benign, malignant and atypical, which are further subtyped into seven categories. Here, since the WSIs number is limited, we only perform three class subtyping.

TCGA [69]: Breast Invasive Carcinoma (BRCA, n = 1, 041, WSI = 1, 111), Colon and Rectum Adenocarcinoma (CRC, n = 566, WSI = 575), Bladder Urothelial Carcinoma (BLCA, n = 373, WSI = 437), Uterine corpus endometrial carcinoma (UCEC, n = 504, WSI = 565), Kidney renal clear cell carcinoma (KIRC, n = 511, WSI = 517), Brain Lower Grade Glioma (LGG) and Glioblastoma Multiforme (GBM) constitute WSI = 463. The train/val split is performed on the patient level.

A.3 Experimental settings

VQ Pretraining: We conduct VQ pretraining on 1M randomly cropped 224×224 tiles extracted from all TCGA [69] diagnostic pathology WSIs. During training, the FM backbone (e.g., UNI with ViT-Large) remains frozen. The input tile images are augmented using `RandomCrop` (minimum ratio: 0.4) and `RandomHorizontalFlip` (probability: 0.5). The codebook has a size of $C = 8192$ with an embedding dimension of 16. The MLP encoder consists of two linear layers with a `tanh` activation in between, transforming the feature dimension from 1024 to 16. The decoder first upsamples the feature dimension from 16 to 768 using a linear layer, followed by three Transformer blocks. Another linear layer then maps the features from 768 to 1024, ensuring alignment with the original feature tokens.

For MSVQ, we employ a multi-scale resolution list: $\{1 \times 1, 2 \times 2, 4 \times 4, 7 \times 7, 14 \times 14\}$.

The model is trained on 4 RTX-3090 GPUs for 50 epochs using a batch size of 128 tile images. The total training time is approximately 22 hours. The learning rate is set to 2×10^{-4} with a 5-epoch warmup, followed by cosine decay to a minimum learning rate of 1×10^{-5}. The weight decay is 1×10^{-4}, and the AdamW optimizer is used with β parameters set to $(0.9, 0.99)$.

WSI-SSL Pretraining: We crop all TCGA diagnostic WSIs into regions of resolution 3584×3584, yielding a dataset of approximately 250k regions. To facilitate SSL, a pretrained MSVQ model is used to extract the quantized indices of each tile within a region, requiring only about 65MB for storage.

All pretraining is conducted on 4 GPUs for 20 epochs with an initial learning rate of 5×10^{-4}. The first 2 epochs serve as a warmup phase, followed by cosine decay to a minimum learning rate of 1×10^{-5}. The AdamW optimizer is employed with β parameters set to $(0.9, 0.98)$ to ensure fast convergence.

For ABMIL pretraining, a batch size of 64 is used due to the model's simplicity. For WSI-Transformer, the batch size is set to 32, with 96 masked tokens out of 256. The learning objective for both models is formulated as a cross-entropy loss over 8192 categories, with ABMIL additionally utilizing soft targets.

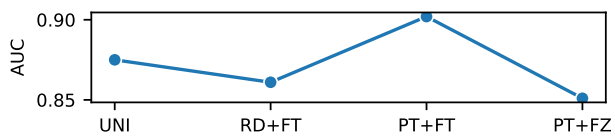

Figure 6: The PathVQ feature with Convs need pretraining and aligning tile's level-0 feature to attain good feature space and compression. RD: random initialize Convs. FT: fine-tuning during WSI analysis. PT: pretrained during VQ learning (align to level-0 tile feature). FZ: freeze during WSI analysis.

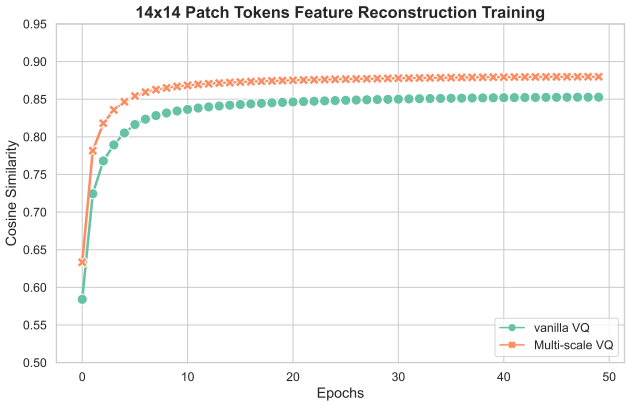

Figure 7: Reconstruction using Multi-Scale (MSVQ) or not. MSVQ obviously improve the rec performance.

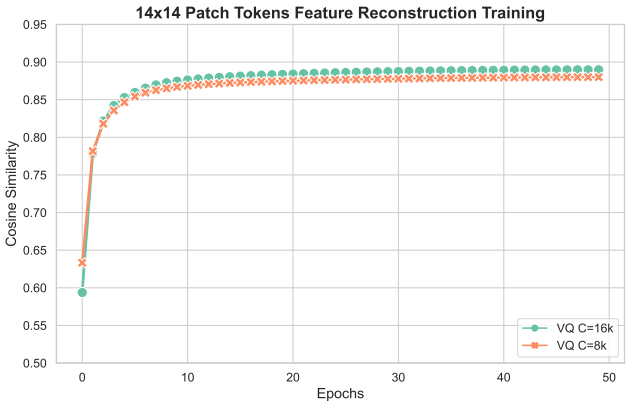

Figure 8: Reconstruction ablation on quantization codebook size, 8k and 16k.

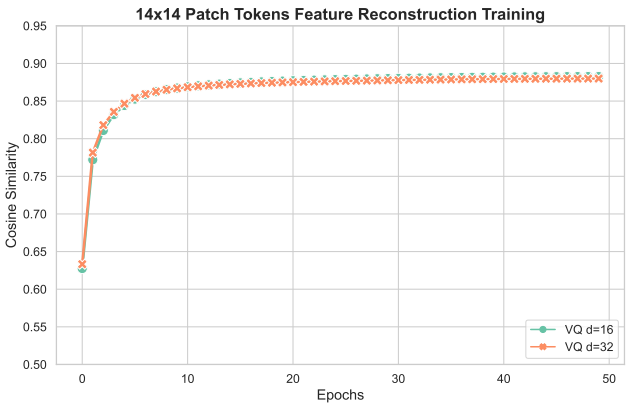

Figure 9: Reconstruction ablation on quantization codebook embedding, 16 and 32.

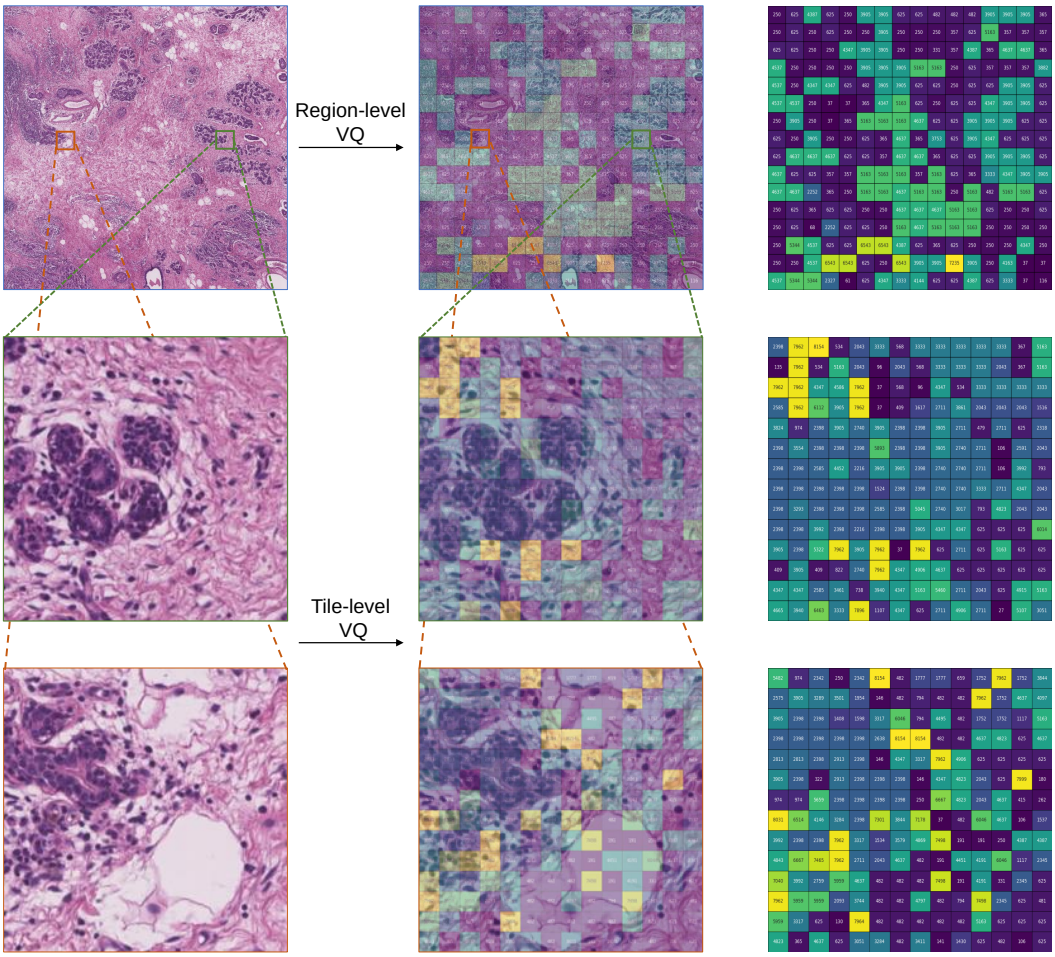

Figure 10: The VQ are performed on both tile-level and patch-level. The quantized index can be seen as a type of prototypes (n=8k) with strong interpretability. Be aware that since the codebook is too large, the heatmap on different index may share similar color.

A.4 Ablations on VQ

Multi-Scale (MSVQ) Please check Figure 7.

Codebook Size Please check Figure 8.

Codebook Embedding Dimension Please check Figure 9.

A.5 Illustrative Visualization

Please check Figure 10.

A.6 Survival Prediction Loss Comparison

Most current MIL methods [31, 61] use NLL loss since WSI batch size is limited (most settings are 1) when the MIL models need fine-tuned with large bag size (GPU memory limit). However, the NLL loss is not optimal for this setting [60]. Cox loss [60], on the other hand, is better than NLL in performance but it need large batch size to calculate the hazard ranking matrix among different samples, which is currently inevitable for MIL models. However, this is easy to be implemented if the bag-size is small (e.g. using unsupervised learning) to prototyping instances' features like PANTHER [63]. Or using a strong WSI pretrained module like TITAN [17] pre-extract the slide-level representations. Though currently our method can not surpass above methods, but it show strong

improvement compared to other baselines which also using NLL loss. And we will further explore this problem (combining Cox loss into fine-tuning WSI-level representation model) in the future.