Weakly Supervised Vector Quantization for Whole Slide Image Classification

Dawei Shen
Division of Health Medical Intelligence
Institute of Medical Science
The University of Tokyo
Tokyo, Japan
coalball@g.ecc.u-tokyo.ac.jp

Yao-zhong Zhang*
Division of Health Medical Intelligence
Institute of Medical Science
The University of Tokyo
Tokyo, Japan
yaozhong@ims.u-tokyo.ac.jp

Seiya Imoto*
Division of Health Medical Intelligence
Institute of Medical Science
The University of Tokyo
Tokyo, Japan
imoto@hgc.jp

***Abstract*—Whole Slide Images (WSIs) are gigapixel, high-resolution digital scans of microscope slides, providing detailed tissue profiles for pathological analysis. Due to their gigapixel size and lack of detailed annotations, Multiple Instance Learning (MIL) becomes the primary technique for WSI analysis. However, current MIL methods for WSIs directly use embeddings extracted by a pretrained vision encoder, which are not task-specific and often exhibit high variability. To address this, we introduce a novel method, VQ-MIL, which maps the embeddings to a discrete space using weakly supervised vector quantization to refine the embeddings and reduce the variability. Additionally, the discrete embeddings from our methods provides clearer visualizations compared to other methods. Our experiments show that VQ-MIL achieves state-of-the-art classification results on two benchmark datasets. The source code is available at https://github.com/aCoalBall/VQMIL.**

***Index Terms*—WSI, Multiple Instance Learning, Vector Quantization**

I. Introduction

Whole Slide Images (WSIs) are high-resolution digital scans of entire microscope slides, widely used in pathology. [1] In recent years, deep learning methods for analyzing these WSIs have advanced rapidly, achieving performance on par with or even exceeding that of pathologists. This progress enables scalable and thorough examination of tissue samples in clinical practice.

One of the key applications of deep learning methods for WSI analysis is WSI tumor classification, which focuses on identifying the presence of tumor tissues within WSIs. This classification task is well-suited for Multiple Instance Learning (MIL)[2, 3], a form of weakly supervised machine learning that organizes data into labeled bags, each containing multiple instances. This method is advantageous for WSI tumor classification for two reasons. First, the gigapixel size of WSIs makes it infeasible for current hardware and algorithms to process them directly. This necessitates dividing each WSI into numerous smaller image tiles and compressing each tile into a small embedding vector. Second, detailed annotations for WSIs are scarce, as providing pixel-level annotations for each slide is costly and time-consuming for pathologists.[4, 5] Hence, training usually utilizes slide-level labels rather than detailed instance labels.

The current MIL process for WSIs begins with a pretrained vision encoder, such as ResNet [6] or vision transformer [7], which compresses each image tile into an embedding vector. These vectors represent the instances in the MIL framework. A MIL model is then trained on these embedding vectors to perform slide classification. This process efficiently addresses the challenges of size and annotation scarcity inherent in WSI tumor classification. A key factor of this process is the quality of the embedding vectors.

However, the embedding vectors obtained from pretrained vision encoders have two drawbacks. The first issue is that the embeddings are not task-specific. The pretraining frameworks are designed to learn general features from a large and diverse dataset, rather than focusing on a specific task like tumor classification. The second issue is the high variability of the continuous embedding vectors. Continuous embeddings from the vision encoder introduce extraneous variations, such as subtle differences in texture or appearance between tiles, impairing the MIL model's ability to perform accurate classification.

To address these two issues, we propose VQ-MIL, which incorporates a weakly supervised vector quantization module between the vision encoder and the MIL model. The module includes a learnable codebook with a fixed number of codewords. During training, embedding vectors from the pretrained vision encoder are refined by an additional encoder to highlight the task-specific information, then these embeddings are mapped to the nearest codewords. These discretized codewords, rather than the original embedding vectors, are passed to the MIL model. This process can be treated as a supervised clustering process that reduces variability of the embeddings and thereby enhancing the MIL model's classification performance. In III-B2, we further explain the effectness of our method using the information bottleneck principle [8]. Additionally, the trained codebook has fixed values, providing clearer and more accurate visualizations compared to continuous embedding vectors used in previous methods[9, 10].

Furthermore, we argue that another bottleneck for WSI classification is the lack of slide-level labels. To address this, we randomly select image tiles from tumor and normal slides

to create positive and negative pseudo-bags as a simple data augmentation method.

We evaluated our method on two representative datasets, Camelyon16 [11] and TCGA lung cancer dataset, and the experimental results show that our model achieves state-of-the-art slide-level performance compared with the previous methods.

To our knowledge, this is the first use of vector quantization in weakly supervised classification. It demonstrates that vector quantization can refine the embedding space for WSI classification, improving diagnostic accuracy.

II. Related Works

A. MIL Methods for WSI Classification

To address the challenges of gigapixel size and the lack of precise annotations in WSIs, current deep learning methods typically employ a two-stage approach that consists of a vision encoder followed by a MIL model. In this process, a WSI is initially divided into many smaller image tiles. A pretrained and frozen vision encoder is then used to convert each of these tiles into one-dimensional embedding vectors. There are two main types of pretrained vision encoders. The first type is a general-purpose vision encoder pretrained on datasets like ImageNet. However, there is a significant domain gap between ImageNet images and pathological images. To address this issue, some research [12, 13, 14, 15] has focused on using a large number of unlabeled WSIs to pretrain vision encoders within self-supervised learning frameworks such as SimCLR [16]or DINO [17, 18].

After extracting the embedding vectors, all vectors from each WSI are collected to train a MIL model that acts as a slide-level classifier. There has been significant work in developing efficient MIL models for WSI classification. AB-MIL [9] introduces a gated-attention mechanism [19] to aggregate features using attention scores. CLAM [10] is a variant of AB-MIL that clusters instances with the top-k highest and lowest attention scores. DS-MIL [12] combines embedding vectors from multiple scales and selects the most representative instance as a query to identify other instances. TransMIL [20] utilizes Transformer[21] architecture to consider the relationships between instances in a slide. DTFD-MIL[22] resamples instances within a slide to increase the number of available labels. MambaMIL[23] utilizes the Selective Scan Space State Sequential Model (Mamba)[24] architecture to process long sequences of instances. All of these methods demonstrate reliable performance in slide-level classification.

Recently, there have been some impressive efforts[25, 26, 27] to attempt to finetune the vision encoders to address the issue that the embeddings generated by the vision encoder are not task-specific. Considering the powerful capabilities of vision encoders, this approach has great potential. However, due to the enormous size of WSIs, it is not feasible to finetune the vision encoder using entire WSIs. To mitigate this issue, some studies[26] finetune the model by assigning the slide-level label to each instance within the slide, while some other studies[25, 27] reduce the input size by sampling instances using methods like K-Means[28] or deep variational information bottleneck[29]. However, these methods inevitably result in incomplete or incorrect data, which can impair the model's performance comparing with fully-supervised finetuning.

B. Vector Quantization

Vector quantization is a technique for compressing data by mapping vectors in high-dimensional space to a finite set of prototype vectors or codebook, effectively reducing the data's dimensionality while preserving its essential features.

Vector Quantized Variational Autoencoder (VQ-VAE)[30] is a variant of variational autoencoder (VAE)[31] that utilizes vector quantization. It encodes input data into discrete latent space by mapping continuous features to the nearest codeword in a learnable codebook. Then this discrete representation is used to generate high-quality reconstructions. VQ-VAE is particularly useful for tasks like image generation and speech synthesis, where capturing discrete underlying structures can improve the model's performance and interpretability.

In our work, unlike VQ-VAE which employing vector quantization to reconstruct input data, we employ it to improve the task-specificity of embeddings, mapping them to a discrete space that aligns better with weakly supervised classification. By integrating this into the MIL pipeline, we aim to improve classification accuracy and instance-level distinguishability.

III. Methods

A. Formulation of MIL Methods for WSI

To formalize the process of WSI deep learning analysis, denote the frozen vision encoder as V_e. For an input WSI X = $\{x_1, x_2, ..., x_k\}$, where k is the total number of tiles and x_i is the ith tile that cropped from the WSI. The set of embedding vectors F = $\{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k\}$ is generated by V_e that $\mathbf{f}_i = V_e(x_i)$, $\mathbf{f}_i \in \mathbb{R}^{1\times N}$, N is the length of $\mathbf{f}_i$. F is regarded as the instances of slide X in the settings of MIL. Then the slide label Y of X, is determined by the latent instance labels $\{y_1, y_2, ..., y_k\}$:

$$Y = \begin{cases} 0 & \text{if } \sum_{i=1}^{k} y_i = 0 \\ 1 & \text{otherwise} \end{cases}$$

Since the labels of instances are unknown, usually a MIL model A is applied on F to produce slide-level prediction $\hat{p} = A(F)$. For gated-attention[19] based MIL models like AB-MIL[9], a learnable attention score a_i is calculated for each instance $\mathbf{f}_i$ in the following way:

$$a_i = \frac{\exp\left(\mathbf{w}^T(\tanh\left(\mathbf{V}\mathbf{f}_i^T\right) \odot \text{sigmoid}\left(\mathbf{U}\mathbf{f}_i^T\right))\right)}{\sum_{j=1}^{N} \exp\left(\mathbf{w}^T(\tanh\left(\mathbf{V}\mathbf{f}_j^T\right) \odot \text{sigmoid}\left(\mathbf{U}\mathbf{f}_j^T\right)\right)}$$

where $\odot$ is the element-wise multiplication operation, $\mathbf{w}$, $\mathbf{U}$ and $\mathbf{V}$ are learnable parameters. $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{M\times N}, \mathbf{w} \in \mathbb{R}^{M\times 1}$, that is, $\mathbf{U}$ and $\mathbf{V}$ linearly map the input from dimension N to M.

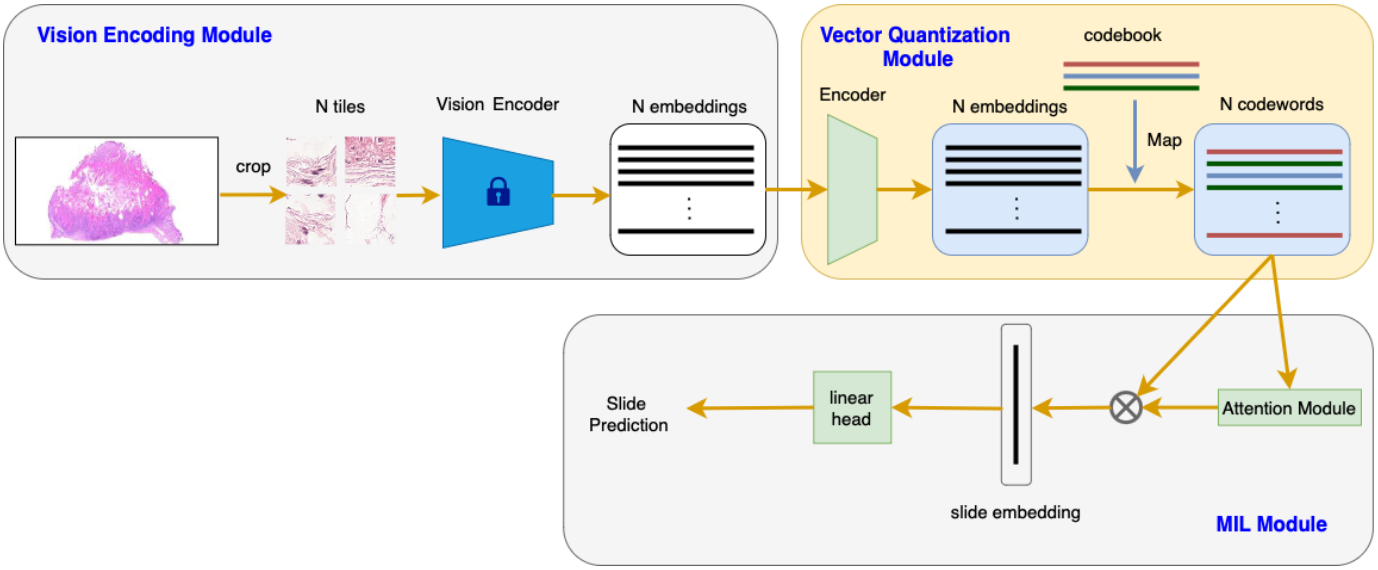


Fig. 1: The workflow of VQ-MIL, which have three modules. The vision encoding module converts a WSI to embedding vectors. The vector quantization module mapped embedding vectors to discrete and task-specific codewords. The MIL module aggregates all codewords to make the prediction.

All instances in a slide are then aggregated as a slide-level embedding and a linear head is used to do the prediction. the model could be described as :

$$A(F) = \sigma(\sum_{i=1}^{N} a_i \mathbf{f}_i)$$

where σ is the linear classifier head.

B. VQ-MIL

In this section we introduce our model VQ-MIL, especially the vector quantization module in it and how it maps the embedding vectors to a discrete and task-specific space. The overview workflow of our method is shown in Figure 1. We split our workflow into three modules, the vision encoding module, the vector quantization module and the MIL module. In the vision encoding module, the input WSI X is first divided into N tiles. A vision encoder V_e independently encodes these N tiles into N embedding vectors. Note that these N embedding vectors are general and not task-specific. They are then passed to the vector quantization module, where a learnable encoder Z_e refines them. As Z_e is learnable, it non-linearly transforms these input embedding vectors, highlighting the task-specific information. Then there is a learnable codebook Q that contains K discretized codewords. All these N embedding vectors are respectively mapped to the nearest codeword. Then it is the mapped codewords instead of the embedding vectors are passed to the MIL module. Here we simply apply the classic AB-MIL as the MIL model A, in which an attention score is calculated for each codeword and all codewords are aggregated as a slide embedding by the weight of attention scores. A linear layer is applied on the slide embedding to do slide prediction.

1) Vector Quantization Module: Inspired by VQ-VAE, we tried to apply vector quantization in MIL to map the continuous embedding vectors to a discrete space. Here, to apply the vector quantization method on WSI classification problems, we add an additional vector quantization module after the conventional vision encoding module and before the MIL module.

In the vector quantization module, a codebook is defined as $Q \in R^{K \times D}$, where K is the size of the codebook and D is the dimension of each codewords. For a WSI X, the generated embedding vectors F are passed through the vector quantization module, where an additional encoder Z_e further encodes the embeddings as E = $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k\}$, where $\mathbf{e}_i = Z_e(\mathbf{f}_i)$. Then the set of codewords C = $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$ is calculated in the way that for each $\mathbf{e}_i$, $\mathbf{c}_i$ is the nearest codeword from it:

$$\mathbf{c}_i = \arg\min_{\mathbf{q}_j} \|\mathbf{e}_i - \mathbf{q}_j\|_2, \forall \mathbf{q}_j \in Q$$

Then it is the calculated set of code words C instead of F is passed through the MIL model A to do the prediction $\hat{p} = A(C)$. C is in a discrete space since it is a set of codewords from the codebook.

Here we face the problem that the vector quantization process is not differentiable. We apply the same method as in VQ-VAE[30] that directly copying the gradient from C to E. Thus the actual expression of the workflow is:

$$\hat{p} = A(sg(C - E) + E)$$

where $E = Z_e(F)$. $sg(x) = x$ during forward propagation and $sg(x) = 0$ during back propagation.

Besides the classification loss, we also add two vector quantization loss similar to which in VQ-VAE that updates

both the outputs of the encoder and the codewords to make them closer.

Therefore the loss function is:

$$L = L_{cls}(\hat{p}, Y) + \|sg(E) - C\| + \alpha\|E - sg(C)\|$$

where Y is the slide label and α is a coefficient. The first term is the slide-level cross-entropy loss, the second and the third term are vector quantization loss. During the training, the parameters of the vector quantization module and the MIL module are updated simultaneously.

Note that compared with VQ-VAE[30], our VQ-MIL replaces the decoder with the MIL model and replaces the reconstruction loss from the decoder with the weakly supervised slide-level classification loss from the MIL model.

Algotithm1 is the pseudo code of the workflow.

Algorithm 1 pseudo-code of VQ-MIL

```
Q ∈ R^{K×D}                                  #the codebook Q
X = {x_1, x_2, ..., x_k}                     #one WSI
V_e.freeze                                   #freeze the vision encoder V_e
F = V_e(X)                                   #get embeddings of X by V_e
E = Z_e(F)                                   #encodes F to E
for e_i ∈ E do
    c_i = arg min_{q_j} ||Z_e(e_i) − q_j||_2, ∀q_j ∈ Q
end for
C = {c_1, c_2, ..., c_k}                     #the set of codewords C
p̂ = A(C)                                     #make the prediction by the MIL model A
ClsLoss = CELoss(p̂, Y)
Loss = ClsLoss + VQLoss
Loss.backpropagate(A, Q, Z_e)                #update Z_e, the codebook
and the MIL model.
```

2) Vector Quantization and Information Bottleneck Principle: Here, we use the information bottleneck principle to explain the effectiveness of vector quantization. In supervised learning, input variable X is often complex and high-dimensional while label variable Y is simple and low-dimensional, meaning much of the information in X is irrelevant to Y [8]. According to Tishby et al. (2015), a key goal of supervised learning is to capture and efficiently represent the features in X that are relevant to Y. Therefore, the information bottleneck principle can be applied to analyze these models. The information bottleneck principle is a theoretical framework trying to find the embedding $\hat{Z}$ that minimizes the following B:

$$B = I(X, \hat{Z}) - \beta I(\hat{Z}, Y)$$

where X is the input variable, Y is the label variable, β is an coefficient and the function I is the mutual information which mathematically measures the amount of information one variable contains about another.

In other words, the information bottleneck principle requires the embedding $\hat{Z}$ to retain more information relevant to Y, but reduce irrelevant information.

In VQ-MIL, the vector quantization module acts as a weakly supervised clustering process. Guided by slide-level labels, embedding vectors are non-linearly clustered around codewords with only the codewords proceeding to the next steps. As a result, $I(X, \hat{Z})$ is implicitly reduced, while $I(\hat{Z}, Y)$ is maintained by the weakly supervised classification loss.

To further illustrate this, we conduct a comparison experiment by replacing the vector quantization module with K-Means (See IV-G3). The results show that vector quantization module effectively reduces irrelevant details, while unsupervised clustering like K-Means only converts general features from continuous to discrete.

C. Random Pseudo Bags

One challenge in training an efficient model for WSI classification is the scarcity of labels. Given the rich and varied content within WSIs, having only one label per slide provides too little information for the MIL model to effectively exploit useful features. For instance, in the Camelyon16 dataset, which contains WSIs of lymph node sections, there are 270 slides for training and 400 slides in total. Usually, thousands of image tiles can be cropped from a single slide, resulting in only 270 labels for hundreds of thousands of tiles in Camelyon16. This lack of labels means that, in gated-attention[19] based MIL models, only a subset of positive tiles may be identified from the slide. This limitation hampers the MIL model's ability to fully learn and distinguish the features of tumor tissues.

Inspired by previous methods[22, 32, 33], we randomly select M instances each time from each slide type (e.g., tumor and normal) to create a large number of pseudo-bags. This approach helps our model explore as many potential instances as possible.

IV. Experiments

A. Datasets and Preprocessing

We report our results on two commonly used WSI datasets: Camelyon16[11] and the TCGA lung cancer dataset.

Camelyon16. The Camelyon16 dataset is a benchmark in computational pathology, used for developing algorithms to detect breast cancer metastases in lymph node tissue sections. It includes 400 whole-slide images with detailed annotations provided by expert pathologists. The dataset is divided into 270 training slides from two medical centers and 130 test slides from another center. This dataset was created for the Camelyon16 Grand Challenge to evaluate the performance of automated systems in improving cancer diagnosis. Camelyon16 also provides pixel-level annotation for each slides for detailed evaluation.

TCGA lung cancer. This dataset includes WSIs for two major subtypes of lung cancer: lung adenocarcinoma (LUAD) and lung squamous cell carcinoma (LUSC). It contains a total of 1,054 slides, with 531 from LUAD and 513 from LUSC. The dataset can be downloaded from the National Cancer Institute Data Portal.

For data preprocessing, we followed the CLAM[10] library guidelines, using the OTSU[34] algorithm for segmentation

TABLE I: Comparison of MIL workflows on Camelyon16. The standard deviations are in parentheses. The best scores are in bold.

Model	Accuracy	AUC
AB-MIL	0.8326 (±0.043)	0.8382 (±0.021)
CLAM	0.8465 (±0.029)	0.8766 (±0.030)
TransMIL	0.8775 (±0.014)	0.8883 (±0.017)
DTFD-MIL	**0.8837** (±0.010)	0.9147 (±0.011)
MambaMIL	0.8605 (±0.024)	0.9106 (±0.007)
Ours	**0.8837** (±0.022)	**0.9248** (±0.014)

TABLE II: Comparison of MIL workflows on TCGA Lung Cancer Dataset. The standard deviations are in parentheses. The best scores are in bold.

Model	Accuracy	AUC
AB-MIL	0.8474 (±0.021)	0.9111 (±0.021)
CLAM	0.8676 (±0.036)	0.9489 (±0.013)
TransMIL	0.8877 (±0.057)	0.9618 (±0.027)
DTFD-MIL	0.8809 (±0.022)	0.9349 (±0.012)
MambaMIL	0.8718(±0.033)	0.9449 (±0.019)
Ours	**0.8884** (±0.016)	**0.9647** (±0.010)

and tile cropping. The tile size was set to 224 × 224 at 20× magnification.

B. Evaluation Metrics

For Camelyon16, the provided training set was split into training and validation sets with a ratio of 0.9:0.1. We conducted five cross-validation experiments for a fair comparison. For the TCGA lung cancer dataset, we randomly split the dataset into training, validation, and test sets with ratios of 0.65, 0.1, and 0.25, respectively. We conducted three cross-validation experiments for this dataset. We report the mean and standard deviation of accuracy and AUC of our method alongside all baselines on both datasets.

C. Implementation Details

For a fair comparison, we use an ImageNet-pretrained ResNet50[6] as the frozen vision encoder to generate embedding vectors for our method and all baselines. It encodes image tiles of size 224 × 224 into embedding vectors of size 1024.

For all baselines, we used the same settings as described in their respective papers and source code.

For our method, we use a 3-layer MLP as the encoder Z_e, which projects the embedding vectors from size 1024 to size 512. We set the number of codewords in the codebook to 32, resulting in a codebook size of 32 × 512. The optimizer used is AdamW[35] with a learning rate of 0.0001. The batch size is set to 1, meaning only one slide is processed per iteration. Due to the different sizes of the datasets, we trained our model for 400 epochs on Camelyon16 and 100 epochs on the TCGA lung cancer dataset. We use pseudo-bags of size 256 as an additional data augmentation technique.

D. Slide-Level Performance

We compare our models with the following baselines: (1) gated-attention[19] based methods, including the classic AB-MIL[9] and its two variants, CLAM[10] and DTFD-MIL[22]; (2) the transformer-based method, TransMIL[20]; and (3) the latest Mamba-bsed method MambaMIL[23]. II-A provides brief introductions to these methods.

Table I summarizes the slide-level performance of the baseline models and our method on Camelyon16. The results show that our method improved accuracy by 5.11% and AUC by 8.66% compared to the classic AB-MIL. Our method also outperforms CLAM and TransMIL. Compared to DTFD-MIL, our method achieved similar accuracy but improved AUC by 1.01%.

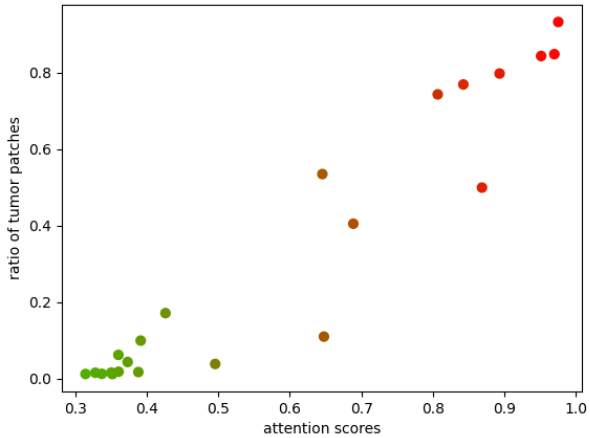
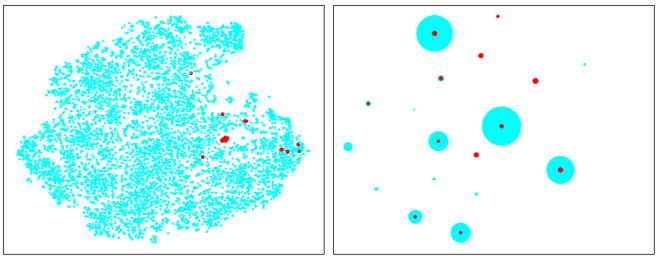


Fig. 2: The relationship between the proportion of tumor instances of each codeword and the attention scores. Codewords with higher attention scores are red, and codewords with lower attention scores are green.

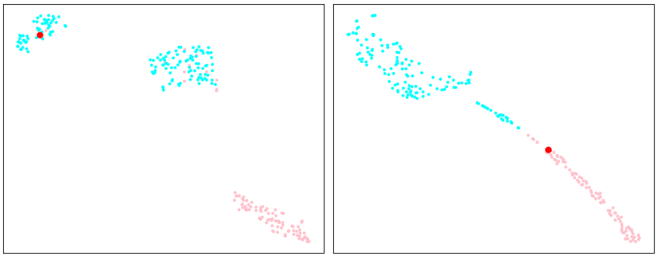
For TCGA lung cancer dataset, table II summarizes the slide-level results. Our method achieved the best performance, with an improvement in accuracy of 4.11% and AUC of 5.36% compared to the classic AB-MIL.

E. Distinguishability between the Tumor Instances and Normal Instances

We further tested the ability of the vector quantization module to reduce variability and distinguish between tumor and normal instances. We calculated the proportion of tumor instances corresponding to the 32 codewords of VQ-MIL trained on Camelyon16 and examined the attention scores associated with these codewords. The results are shown in Figure 2. We can see that different codewords have vastly different proportions of tumor instances. Some codewords contain more than 90% tumor instances, while many others consist almost entirely of normal instances. This indicates that codewords can effectively provide the task-specific information like distinguishing between tumor and normal instances. Additionally, there is a strong correlation between the propor-



(a) T-SNE plot of embedding vectors of a WSI (tumor_093) in Camelyon16. The red points represent tumor instances and the cyan are normal. Left figure is the embedding vectors by the pretrained ResNet50. Right figure is the mapped discrete codewords after vector quantization, where the size of each point represents the number of instances that are mapped to this codeword. We can see that after the mapping, some tumor insances are separated from the normal ones.



(b) T-SNE plot of the aggregated slide embedding vectors of the training set of Camelyon16. Left figure is the result of AB-MIL. Right figure is the result of our proposed method. The cyan points represent normal slides and the pink are tumor slides. The red point is the WSI in (a), we can see that the AB-MIL cannot predict this WSI correctly while our proposed method successfully do the classification.

Fig. 3: The t-SNE plot of embedding vectors in Camelyon16.

tion of tumor instances corresponding to a codeword and its attention score. Codewords with a high proportion of tumor instances correspond to high attention scores, whereas those with a low proportion correspond to low attention scores. This suggests that the model could effectively notice the codewords with higher tumor proportions for classification prediction.

Figure 3a and 3b show the t-SNE plots of instance embedding vectors from a WSI in Camelyon16 and the slide embedding vectors obtained from all WSIs in the training set, respectively. In 3a, we observe that the instance embedding vectors generated by the vision encoder contain significant noise, with tumor instances buried among normal instances, making them difficult to distinguish. However, after being discretely mapped to codewords by the vector quantization module, some tumor instances are separated from the normal instances. This separation allows the MIL model to notice these distinct tumor instances, leading to stronger distinguishability in the slide embedding vectors, as reflected in 3b.

F. Instance-Level Annotation and Visualization

In this section, we demonstrate that our method provides better visualization for instance-level annotation of WSIs compared to previous models. Currently, drawing heatmaps of attention scores using gated-attention[19] based MIL models is the common method for instance-level visualization in WSI classification. Compared to the previous gated-attention[19] based models, our method has an advantage for visualization. In our model, the attention scores derived from the codewords are fixed. In most models, attention scores has to be normalized in the model so that the sum of the attention scores for all instances in a slide is 1. This process can be detrimental to visualization. For example, higher attention scores usually should indicate a higher likelihood of being a tumor instance. However, due to normalization, there is no clear threshold for this likelihood, even normal slides may have many instances with high attention scores. In our model, although normalization is applied during prediction, the codebook is fixed after training. Thus we can use this codebook as a bag with K instances (codewords) and input it into the MIL model to obtain fixed attention scores. This approach results in better visualization outcomes. Figure 4 illustrates the difference in visualization performance.

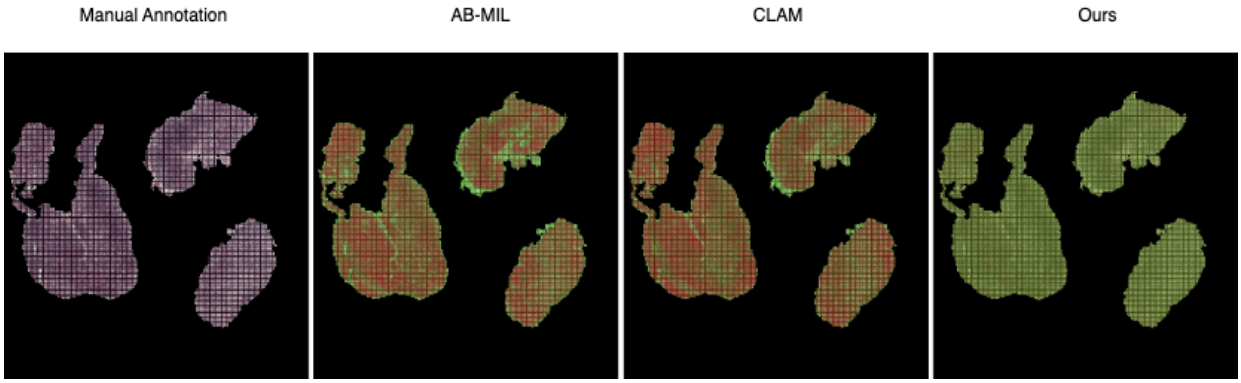
G. Ablation Study

1) Contribution of each component: We tested the three components of our VQ-MIL:the encoder Z_e, the vector quantization process, and the data augmentation through pseudo bags.

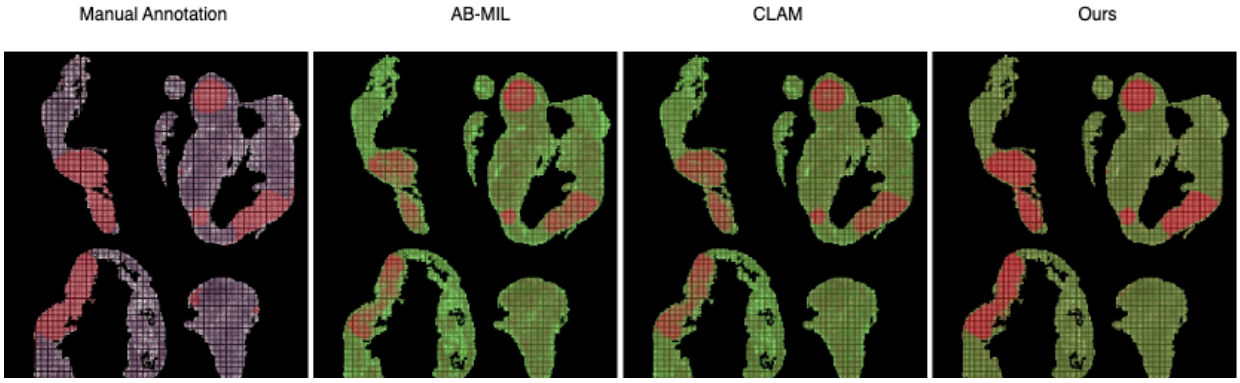
Table III summarize the contribution of each component on Camelyon16. In the table, the first row represents the basic AB-MIL model, which shows the lowest performance. The second row features AB-MIL with an additional encoder Z_e. The encoder Z_e encodes the embedding vectors to be more task-specific, although there is no vector quantization process to map the vectors to a discrete space. This addition improved the AUC by 2.77% and accuracy by 2.96% compared to the basic AB-MIL, indicating the importance of converting embedding vectors to be more task-specific. The third row represents VQ-MIL without additional data augmentation. It further improved the AUC by nearly 3% and accuracy by 1.24%, suggesting that a discrete embedding space is beneficial. The last row is the full version of our method, which achieves the best performance, highlighting the contribution of data augmentation through pseudo-bags. In conclusion, we find that each component significantly contributes to the overall performance.

2) Impact of different codebook size: Table IV shows the slide-level performance on the Camelyon16 dataset concerning different codebook sizes. For binary classification tasks, it might seem intuitive to set the codebook size to two; however, the performance is unsatisfactory because the encoder of the vector quantization module struggles to accurately cluster all instances into just two codewords. Increasing the size of the codebook improves performance. We also find that there is almost no difference in performance when the codebook size exceeds 16.

3) Comparison with unsupervised clustering: To highlight the advantages of the weakly supervised vector quantization module, we conduct a comparison experiment by replacing



(a) The visualization of a normal WSI in Camelyon16



(b) The visualization of a tumor WSI in Camelyon16

Fig. 4: Two visualization examples in Camelyon16. The left-most column is the provided ground truth annotation. The second column is the heatmap from the classic AB-MIL. The thrid column is the heatmap from CLAM. The right-most column is the heatmap from our method.

it with K-Means, which is a type of unsupervised clustering. In this setup, each embedding vector is replaced by its corresponding cluster centroid. As an unsupervised algorithm, K-Means is not task-specific and merely converts general embeddings from continuous to discrete. Table V shows the results of K-Means on Camelyon16 under different value of K. It yields AUC that is about 20% lower and the accuracy is about 15% lower compared to the vector quantization module.

4) Impact of different pseudo-bag size: Table VI shows the slide-level performance on Camelyon16 concerning different sizes of pseudo-bags. When the pseudo-bag size is too small, performance decreases rapidly. Since slides in Camelyon16 usually contain only a tiny fraction of tumor tissues, a small pseudo-bag may not include any positive instances. The best performance is achieved when the pseudo-bag size is 256. Performance decreases when the size of the pseudo-bags becomes larger because larger bags contain many simple instances, which are not helpful for the model to identify challenging instances.

TABLE III: Contribution of different components of VQ-MIL on Camelyon16. Best scores are in bold.

	Accuracy	AUC
AB-MIL	0.8326	0.8382
Encoder+AB-MIL	0.8682	0.8659
VQ+AB-MIL	0.8806	0.8955
VQ+AB-MIL+pseudo-bags (ours)	**0.8837**	**0.9248**

TABLE IV: Different size of the codebook on Camelyon16

size	Accuracy	AUC
2	0.8465	0.8432
8	0.8202	0.7842
16	0.8620	0.9080
32	0.8806	0.8955
64	0.8806	0.8885
128	0.8744	0.8881
256	0.8806	0.9002
512	0.8760	0.8954

V. Conclusion and Future Works

In this paper, we proposed VQ-MIL, a novel method that integrates weakly supervised vector quantization into the MIL framework for WSI classification. By transforming continuous embeddings into discrete, task-specific codewords, VQ-MIL addresses two challenges in WSI classification, the lack of task-specificity and the high variability associated with con-

TABLE V: KMeans (Unsupervised Clustering) Results on Camelyon16.

K	Accuracy	AUC
2	0.6946	0.6976
8	0.7054	0.7040
16	0.6791	0.6438
32	0.7318	0.6944
64	0.7333	0.6887
128	0.7318	0.6982

TABLE VI: Different size of the pseudo bags on Camelyon16.

size	Accuracy	AUC
128	0.8031	0.8119
256	0.8837	0.9248
512	0.8589	0.8859
8192	0.8558	0.8294

tinuous embeddings. Our experiments demonstrate that VQ-MIL not only achieves state-of-the-art classification results on the Camelyon16 and TCGA lung cancer dataset but also improves the interpretability by providing clearer instance-level annotations.

Beyond pathology, we believe that the idea of VQ-MIL introduced in our work has broader applications in other weakly supervised tasks where handling large-scale data with limited annotations is common.

In future work, we plan to explore the potential of extending VQ-MIL's discretization approach to more granular instance-level classification. We also aim to investigate optimizing the codebook design and exploring other vector quantization techniques to further enhance model performance and applicability.

Acknowledgment

The computing resources in this work were provided by Human Genome Center, the Institute of Medical Science, the University of Tokyo.

We would like to thank Yohei Okubo and Yuxuan Pang for their valuable assistance and contributions to this research.

References

[1] T. C. Cornish, R. E. Swapp, and K. J. Kaplan, "Whole-slide imaging: Routine pathologic diagnosis," *Advances in Anatomic Pathology*, vol. 19, no. 3, 2012.

[2] L. Qu, S. Liu, X. Liu, M. Wang, and Z. Song, "Towards label-efficient automatic diagnosis and analysis: a comprehensive survey of advanced deep learning-based weakly-supervised, semi-supervised and self-supervised techniques in histopathological image analysis," *Physics in Medicine & Biology*, vol. 67, no. 20, p. 20TR01, Oct. 2022. [Online]. Available: http://dx.doi.org/10.1088/1361-6560/ac910a

[3] C. L. Srinidhi, O. Ciga, and A. L. Martel, "Deep neural network models for computational histopathology: A survey," *Medical Image Analysis*, vol. 67, p. 101813, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1016/j.media.2020.101813

[4] J. Rony, S. Belharbi, J. Dolz, I. Ben Ayed, L. McCaffrey, and E. Granger, "Deep weakly-supervised learning methods for classification and localization in histology images: A survey," *Machine Learning for Biomedical Imaging*, vol. 2, no. March 2023, p. 96–150, Mar. 2023. [Online]. Available: http://dx.doi.org/10.59275/j.melba.2023-5g54

[5] V. Cheplygina, M. de Bruijne, and J. P. Pluim, "Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Medical Image Analysis*, vol. 54, pp. 280–296, 2019.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[8] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," 2015. [Online]. Available: https://arxiv.org/abs/1503.02406

[9] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," 2018. [Online]. Available: https://arxiv.org/abs/1802.04712

[10] M. Y. Lu, D. F. K. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, "Data efficient and weakly supervised computational pathology on whole slide images," 2020. [Online]. Available: https://arxiv.org/abs/2004.09666

[11] B. Ehteshami Bejnordi, M. Veta, P. Johannes van Diest, B. van Ginneken, N. Karssemeijer, G. Litjens, J. A. W. M. van der Laak, , and the CAMELYON16 Consortium, "Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer," *JAMA*, vol. 318, no. 22, pp. 2199–2210, 12 2017. [Online]. Available: https://doi.org/10.1001/jama.2017.14585

[12] B. Li, Y. Li, and K. W. Eliceiri, "Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning," 2021. [Online]. Available: https://arxiv.org/abs/2011.08939

[13] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, R. G. Krishnan, and F. Mahmood, "Scaling vision transformers to gigapixel images via hierarchical self-supervised learning," 2022. [Online]. Available: https://arxiv.org/abs/2206.02647

[14] H. Xu, N. Usuyama, J. Bagga, S. Zhang, R. Rao, T. Naumann, C. Wong, Z. Gero, J. González, Y. Gu, Y. Xu, M. Wei, W. Wang, S. Ma, F. Wei, J. Yang, C. Li, J. Gao, J. Rosemon, T. Bower, S. Lee, R. Weerasinghe, B. J. Wright, A. Robicsek, B. Piening, C. Bifulco,

S. Wang, and H. Poon, "A whole-slide foundation model for digital pathology from real-world data," *Nature*, vol. 630, no. 8015, pp. 181–188, Jun 2024. [Online]. Available: https://doi.org/10.1038/s41586-024-07441-w

[15] X. Wang, S. Yang, J. Zhang, M. Wang, J. Zhang, W. Yang, J. Huang, and X. Han, "Transformer-based unsupervised contrastive learning for histopathological image classification," *Medical Image Analysis*, vol. 81, p. 102559, 2022.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[17] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021. [Online]. Available: https://arxiv.org/abs/2104.14294

[18] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2024. [Online]. Available: https://arxiv.org/abs/2304.07193

[19] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," 2017. [Online]. Available: https://arxiv.org/abs/1612.08083

[20] Z. Shao, H. Bian, Y. Chen, Y. Wang, J. Zhang, X. Ji, and Y. Zhang, "Transmil: Transformer based correlated multiple instance learning for whole slide image classification," 2021. [Online]. Available: https://arxiv.org/abs/2106.00908

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[22] H. Zhang, Y. Meng, Y. Zhao, Y. Qiao, X. Yang, S. E. Coupland, and Y. Zheng, "Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification," 2022. [Online]. Available: https://arxiv.org/abs/2203.12081

[23] S. Yang, Y. Wang, and H. Chen, "Mambamil: Enhancing long sequence modeling with sequence reordering in computational pathology," 2024. [Online]. Available: https://arxiv.org/abs/2403.06800

[24] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2024. [Online]. Available: https://arxiv.org/abs/2312.00752

[25] Y. Sharma, A. Shrivastava, L. Ehsan, C. A. Moskaluk, S. Syed, and D. E. Brown, "Cluster-to-conquer: A framework for end-to-end multi-instance learning for whole slide image classification," 2021. [Online]. Available: https://arxiv.org/abs/2103.10626

[26] Y. Zhou, S. Che, F. Lu, S. Liu, Z. Yan, J. Wei, Y. Li, X. Ding, and Y. Lu, "Iterative multiple instance learning for weakly annotated whole slide image classification," *Physics in Medicine & Biology*, vol. 68, no. 15, p. 155007, jul 2023. [Online]. Available: https://dx.doi.org/10.1088/1361-6560/acde3f

[27] H. Li, C. Zhu, Y. Zhang, Y. Sun, Z. Shui, W. Kuang, S. Zheng, and L. Yang, "Task-specific fine-tuning via variational information bottleneck for weakly-supervised pathology whole slide image classification," 2023. [Online]. Available: https://arxiv.org/abs/2303.08446

[28] J. Wu, *Advances in K-means clustering: a data mining thinking*. Springer Science & Business Media, 2012.

[29] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," 2019. [Online]. Available: https://arxiv.org/abs/1612.00410

[30] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2018. [Online]. Available: https://arxiv.org/abs/1711.00937

[31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: https://arxiv.org/abs/1312.6114

[32] P. Liu, L. Ji, X. Zhang, and F. Ye, "Pseudo-bag mixup augmentation for multiple instance learning-based whole slide image classification," *IEEE Transactions on Medical Imaging*, vol. 43, no. 5, p. 1841–1852, May 2024. [Online]. Available: http://dx.doi.org/10.1109/TMI.2024.3351213

[33] A. Tarkhan, T. K. Nguyen, N. Simon, and J. Dai, "Investigation of training multiple instance learning networks with instance sampling," in *Resource-Efficient Medical Image Analysis*, X. Xu, X. Li, D. Mahapatra, L. Cheng, C. Petitjean, and H. Fu, Eds. Cham: Springer Nature Switzerland, 2022, pp. 95–104.

[34] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019. [Online]. Available: https://arxiv.org/abs/1711.05101