Flow-MIL: Constructing Highly-expressive Latent Feature Space For Whole Slide Image Classification Using Normalizing Flow

Yingfan Ma[3,1,2], Bohan An[1,2], Ao Shen[1,2], Mingzhi Yuan[4], Minghong Duan[1,2], Manning Wang[1,2,*]

[1] Digital Medical Research Center, School of Basic Medical Sciences, Fudan University, Shanghai 200032, China

[2] Shanghai Key Laboratory of Medical Imaging Computing and Computer Assisted Intervention, Shanghai 200032, China

[3] Ant Group

[4] Jiangsu Provincial Key Laboratory of Intelligent Medical Image Computing (iMIC), NUIST, Nanjing 210044, China

1. Pseudocode of Flow-MIL

Algorithm 1 gives the details of Flow-MIL.

2. Dataset Description

To demonstrate the performance of the proposed Flow-MIL and compare it to SOTA algorithms, various experiments were conducted over three public datasets: CAMELYON16 [1], TCGA NSCLC [1], PANDA [2]. It's noted that CAMELYON and TCGA-NSCLC datasets are binary classification problems, while the PANDA dataset is a multi-class dataset that contains 3 Gleason pattern grades and instance-level annotations.

2.1. CAMELYON16

CAMELYON is a publicly available dataset for metastasis detection in breast cancer. It consists of 400 H&E-stained WSIs of lymph nodes, of which 270 are used for training and 130 for testing. WSIs containing metastasis are labeled positive, while those without metastasis are labeled negative. The dataset also provides pixel-level labels for metastasis areas. Prior to training, we divided each WSI into non-overlapping 512×512 image patches under 10× magnification.

2.2. PANDA

The Prostate cANcer graDe Assessment (PANDA) is a publicly available dataset designed for diagnosing prostate cancer. Unlike traditional cancer classification methods that focus on binary categorization of cancerous and normal tissue, the PANDA dataset further categorizes cancerous regions into three Gleason pattern grades (3, 4, or 5) based on the tumor's architectural growth patterns. Individual WSI can contain multiple different Gleason pattern regions. The PANDA dataset comprises 10,616 WSIs from two medical centers, of which 5,160 WSIs from Radboud University Medical Center have pixel-level annotations for the Gleason pattern regions. For a balanced distribution of training and validation data, the 5,160 WSIs are split at random: 70% designated for training and the remainder for testing. And, aligning with the processing techniques of the CAMELYON 16 competition, we adopted a uniform approach—slicing the WSIs into patches of 224x224 dimensions at 10x magnification.

2.3. TCGA-NSCLC

TCGA-NSCLC includes 1054 WSIs with two subtypes, i.e., Lung Squamous Cell Carcinoma (TGCA-LUSC) and Lung Adenocarcinoma (TCGA-LUAD), including 1054 WSIs (840 training slides and 210 testing slides) of two subtypes of lung cancer, each slide containing more than 80% tumor area, and the task is to classify these tumor slides into one of the two subtypes. Only slide-level labels are available for this dataset. As the dataset is much easier than CAMELYON16, we use only the pre-extracted instance features provided by instead of end-to-end training.

3. Implementation Details

For feature extraction, we employ a ResNet18 encoder [3] to process image patches. Each feature embedding extracted by the encoder has an initial dimension of 1024, which is subsequently reduced to 512. In the dual-branch structure, the projectors are implemented as linear layers with Batch Normalization, and during training, the two projectors are updated alternately. All experiments are conducted on NVIDIA GPUs. For the CAMELYON16 and PANDA dataset, we extract patches directly from WSIs, while for the TCGA dataset, we utilize the pre-extracted instance features provided by DSMIL [5].

Flow-MIL consists of N $Flow$ blocks. Specifically, for

*Corresponding author.

[1] http://www.cancer.gov/tcga

Algorithm 1 WSI Classification with Flow-MIL

Input: WSI dataset $D = \{(X, Y)\}$, where X are whole slide images embeddings and Y are bag-level labels
Data: Invertible Flow $f(\cdot;\theta_f)$ and Inverse Flow $f^{-1}(\cdot;\theta_f)$, MIL block $M(\cdot;\theta_m)$, instance classifier $I(\cdot;\theta_i)$
Output: Optimized parameters $\theta_f, \theta_m, \theta_i, G$
foreach *training epoch* **do**

Latent Feature Mapping Phase

$Z \leftarrow f(X;\theta_f)$ # Transform WSI patch features into LSES
$\hat{X} \leftarrow f^{-1}(Z;\theta_f)$ # Reconstruct input features to enforce information preservation
$\mathcal{L}_{\text{rec}} \leftarrow \text{MSE}(X, \hat{X})$ # Reconstruction loss ensures invertibility
$P_{\text{latent}}(z_i) \leftarrow \text{GMM}(z_i), \forall z_i \in Z$ # Model latent features with GMM
$\mathcal{L}_{\text{proto}} \leftarrow \text{CrossEntropy}(\bar{P}_{\text{latent}}, Y)$ # Align latent features with bag-level labels

MIL Block Phase

$a_i \leftarrow M(x_i;\theta_m), \forall x_i \in X$ # Compute attention scores for each instance
$X_{\text{agg}} \leftarrow \sum a_i \cdot x_i$ # Aggregate instance features using attention weights
$\hat{Y} \leftarrow \text{Softmax}(X_{\text{agg}})$ # Predict bag label
$\mathcal{L}_{\text{bag}} \leftarrow \text{CrossEntropy}(\hat{Y}, Y)$ # Compute bag-level classification loss

Instance-Level Classification Phase

$y_i^{\text{pseudo}} \leftarrow \lambda_1 \cdot P_{\text{latent}}(z_i) + \lambda_2 \cdot a_i$ # Generate pseudo-labels for instances
$\hat{y}_i \leftarrow I(z_i;\theta_i), \forall z_i \in Z$ # Predict instance-level labels
$\mathcal{L}_{\text{ins}} \leftarrow \sum_i \text{CrossEntropy}(\hat{y}_i, y_i^{\text{pseudo}})$

Optimization Phase

$\theta_f \leftarrow \theta_f - \eta_f \nabla_{\theta_f}(\mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{proto}})$ # Update latent embedding model
$\theta_m \leftarrow \theta_m - \eta_m \nabla_{\theta_m}(\mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{proto}} + \mathcal{L}_{\text{bag}})$ # Update MIL block parameters
$\theta_i \leftarrow \theta_i - \eta_i \nabla_{\theta_i} \mathcal{L}_{\text{ins}}$ # Update instance classifier

the CAMELYON16 and TCGA datasets, we use $N = 5$ flow blocks, while for the PANDA dataset, we use $N = 2$ flow blocks. The GMM used in the latent space has $k = 5$ components, determined based on ablation experiments. Additionally, the MIL block follows the structure of the traditional ABMIL [4] method.

The training process employs the cross-entropy loss and mse loss as the primary objective. For optimization, we use SGD optimizer with a learning rate of 1×10^{-3}.

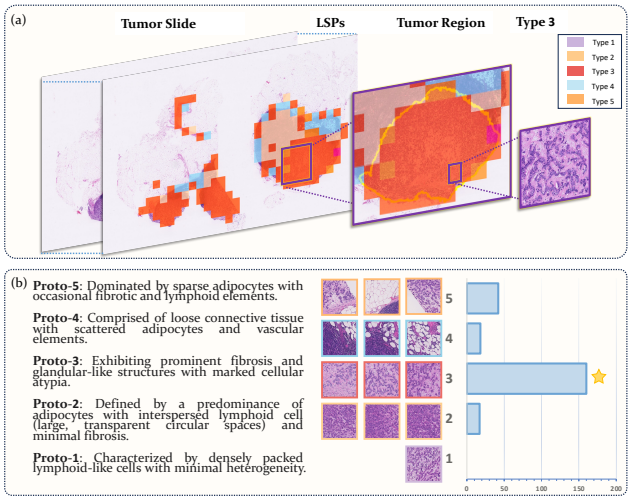

Figure 1. Additional visualization of LSPs for Positive Lymph Node Metastasis. The prototypes in the latent space are visualized for the positive lymph node metastasis category, with instances assigned to the prototype with the highest posterior probability. Color-coded regions on the WSI highlight key pathological features, such as Proto-3 (malignant transformation) demonstrating the latent space captures subtle morphological differences.

4. More Details on Flow-MIL

4.1. More Visualization on CAMELYON dataset

We present additional interpretability visualizations, as shown in Figure 1 from another perspective, further validate how different types of prototypes within the LSP effectively model the positive lymph node metastasis category. These visualizations align with the figures in the main text, showcasing the distinct pathological features captured by each prototype and reinforcing the interpretability of the proposed method.

4.2. Hyper Parameter on Guidance of LSP

To simplify the experimental design, reduce hyperparameter complexity, and enhance model stability, we simplified the LSP guidance mechanism into a single control factor, h. The new simplified formula is:

$$P_{ins} = \frac{h \cdot p(y = c|z_i) + a_i}{1 + h}, \tag{1}$$

where h determines the relative importance of $p(y = c|z_i)$ (classification probability based on latent features) and a_i is attention score obtained by MIL Block.

To evaluate the impact of h, we conducted an ablation study varying h from 0.1 to 0.6. The results, as shown in Table 1, demonstrate that setting $h = 0.2$ achieves the best performance for both instance-level and bag-level AUC. This finding highlights the effectiveness of the simplified

h	Instance-level AUC	Bag-level AUC
0.1	0.9209	0.9176
0.2	**0.9277**	**0.9437**
0.3	0.9230	0.9348
0.4	0.9254	0.9289
0.5	0.9235	0.9277
0.6	0.9054	0.9248

Table 1. Performance comparison across different h values.

guidance approach and its contribution to improving the model's classification accuracy.

4.3. Supplementary Derivation

In this section, we derive the $P(y = c \mid z_i)$, which is fundamental to both instance-level and bag-level classification in our framework. The derivation leverages Gaussian Mixture Model (GMM)-based likelihood modeling and Bayes' theorem. While our primary formulation includes the posterior $P(y = c \mid z_i)$ as a component, its mathematical origin is critical to understanding its role in the framework and was omitted in the main text. Here, we provide the detailed explanation.

The posterior probability $P(y = c \mid z_i)$ is computed using Bayes' theorem as:

$$P(y = c \mid z_i) = \frac{P(z_i \mid y = c)P(y = c)}{P(z_i)}, \tag{2}$$

where, $P(z_i \mid y = c)$ is the likelihood, representing the conditional probability of z_i given class c. $P(y = c)$ is the prior probability of class c. $P(z_i)$ is the marginal probability, obtained by summing over all possible classes. Assuming uniform class priors, i.e., $P(y = c) = \frac{1}{C}$, the equation simplifies to:

$$P(y = c \mid z_i) = \frac{P(z_i \mid y = c)}{\sum_{c'=1}^{C} P(z_i \mid y = c')}. \tag{3}$$

To model $P(z_i \mid y = c)$, we assume a GMM for each class c, expressed as:

$$P(z_i \mid y = c) = \sum_{j=1}^{k} \pi_{c,j} \cdot \mathcal{N}(z_i \mid \mu_{c,j}, \Sigma_{c,j}), \tag{4}$$

where, $\pi_{c,j}$ is the mixing coefficient for the j-th Gaussian component in class c. $\mathcal{N}(z_i \mid \mu_{c,j}, \Sigma_{c,j})$ is the Gaussian distribution with mean $\mu_{c,j}$ and covariance $\Sigma_{c,j}$, and k is the number of Gaussian components per class.

For each Gaussian component j of class c, the posterior probability $p(z_i \mid c, j)$ that z_i belongs to component j within class c is given by:

$$p(z_i \mid c, j) = \frac{\pi_{c,j} \cdot \mathcal{N}(z_i \mid \mu_{c,j}, \Sigma_{c,j})}{\sum_{l=1}^{k} \pi_{c,l} \cdot \mathcal{N}(z_i \mid \mu_{c,l}, \Sigma_{c,l})}. \tag{5}$$

This posterior represents the normalized contribution of the j-th Gaussian component to the total likelihood of z_i under class c.

Substituting $P(z_i \mid y = c)$ into Bayes' theorem, the posterior probability $P(y = c \mid z_i)$ becomes:

$$P(y = c \mid z_i) = \frac{\sum_{j=1}^{k} \pi_{c,j} \cdot \mathcal{N}(z_i \mid \mu_{c,j}, \Sigma_{c,j})}{\sum_{c'=1}^{C} \sum_{j=1}^{k} \pi_{c',j} \cdot \mathcal{N}(z_i \mid \mu_{c',j}, \Sigma_{c',j})}. \tag{6}$$

This formulation ensures that the posterior probability $P(y = c \mid z_i)$ is normalized across all classes, reflecting the relative likelihood of z_i belonging to class c.

With the $P(y = c \mid z_i)$ computed for each instance z_i in a bag, we aggregate these probabilities to estimate the bag-level probability:

$$\overline{P}(y = c \mid \text{Bag}) = \frac{1}{n} \sum_{i=1}^{n} P(y = c \mid z_i), \tag{7}$$

where n is the number of instances in the bag.

4.4. Pipeline of $Flow$ Block and $Flow^{-1}$ Block

Invertible neural networks (INNs) are a special class of network architectures that enable efficient mappings from input space to latent space using carefully designed transformations $f(x)$. These networks not only perform forward mapping but also allow the original input to be perfectly reconstructed through the inverse transformation $f^{-1}(z)$. This design ensures that INNs excel in lossless information processing and density estimation.

The key idea of INNs is to decompose complex mappings into simple, efficient local operations through splitting and coupling transformations. Specifically, input features X are split into two parts, X^A and X^B. The coupling network generates transformation parameters $s(X^A)$ and $t(X^A)$, which are used to perform conditional affine transformations on X^B. This design enables direct computation of the inverse transformation without requiring complex numerical optimization and efficient computation of the Jacobian determinant for density estimation.

The modular design of INNs allows stacking multiple flow blocks to gradually capture the complexity of the input distribution. To enhance stability, flow blocks often include feature permutations or linear transformations to improve network expressiveness.

Algorithm 2 $Flow$ Block Transformations

Require: Input tensor $x \in \mathbb{R}^d$, Coupling network $\theta(\cdot)$
Ensure: Transformed output $z \in \mathbb{R}^d$ and log-determinant of Jacobian $\log|\det J_f(x)|$

1: **Split** x into $x^A \in \mathbb{R}^m$ and $x^B \in \mathbb{R}^{d-m}$.
2: **Compute coupling parameters** $(s,t) = \theta(x^A)$, where $s,t \in \mathbb{R}^{d-m}$.
3: **Affine transformation:**

$$z^A = x^A, \quad z^B = x^B \odot \exp(s) + t$$

4: **Concatenate:** $z = \text{Concat}(z^A, z^B)$.
5: **Compute log-determinant of Jacobian:**

$$\log|\det J_f(x)| = \sum_{i=1}^{d-m} s_i$$

6: **return** $z, \log|\det J_f(x)|$

Algorithm 3 $Flow^{-1}$ Block Transformations

Require: Transformed tensor $z \in \mathbb{R}^d$, Coupling network $\theta(\cdot)$
Ensure: Reconstructed input $x \in \mathbb{R}^d$

1: **Split** z into $z^A \in \mathbb{R}^m$ and $z^B \in \mathbb{R}^{d-m}$.
2: **Compute coupling parameters** $(s,t) = \theta(z^A)$, where $s,t \in \mathbb{R}^{d-m}$.
3: **Inverse affine transformation:**

$$x^A = z^A, \quad x^B = (z^B - t) \odot \exp(-s)$$

4: **Concatenate:** $x = \text{Concat}(x^A, x^B)$.
5: **return** x

References

[1] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22): 2199–2210, 2017. 1

[2] Wouter Bulten, Kimmo Kartasalo, Po-Hsuan Cameron Chen, Peter Ström, Hans Pinckaers, Kunal Nagpal, Yuannan Cai, David F Steiner, Hester van Boven, Robert Vink, et al. Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge. *Nature medicine*, 28(1):154–163, 2022. 1

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607. PMLR, 2020. 1

[4] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning (ICML)*, pages 2127–2136. PMLR, 2018. 2

[5] 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 (CVPR)*, pages 14318–14328, 2021. 1