Enhancing Digital Pathology Visual Understanding With Sparse Pyramid Attention Networks

Anonymous ICCV submission

Paper ID

Abstract

Whole slide image analysis plays a vital role in modern digital pathology, enabling computer-aided diagnosis and advancing research in computational pathology. However, the gigapixel-scale resolutions and sparse informative regions of WSIs pose significant computational challenges. Conventional patch-based methods struggle to model inter-patch relationships accurately, potentially leading to distortions or neglect of important spatial and contextual information. Traditional dense attention mechanisms, while effective in capturing such relationships, become impractical for WSI analysis due to redundant processing of uninformative areas. To address these challenges, we propose Sparse Pyramid Attention Networks, a novel framework consisting of two key modules: Spatial-Adaptive Feature Condensation (SAC) and Context-Aware Feature Refinement (CAR). SAC progressively condenses informative regions into hierarchical multi-scale representations, while CAR comprehensively models both local and long-range contextual relationships via a shifted-window scheme and global context tokens. Extensive experiments on public WSI datasets demonstrate SPAN's performance improvements over state-of-the-art methods, highlighting its potential for advancing computational pathology workflows.

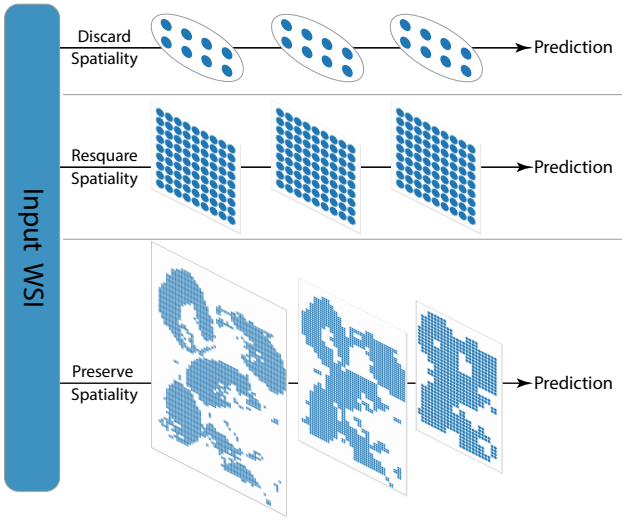

Figure 1. Comparison of our proposed approach with conventional patch-based methods. Top: Methods treat patches as independent and identically distributed (i.i.d.) samples, ignoring the rich spatial structure present in the data. Middle: Approaches reshape patches to a square to encode spatial information. Bottom: Our method constructs a hierarchical representation that captures undistorted spatial relationships and multi-scale contextual information.

1. Introduction

Whole Slide Images (WSIs) have become indispensable tools in modern digital pathology. These high-resolution scans are typically obtained from Hematoxylin and Eosin (H&E)-stained tissue samples, where H&E staining highlights different tissue components for precise identification of cellular structures and abnormalities. The digitization of histopathological slides through WSIs enables pathologists to view and analyze tissue samples at multiple scales, providing a platform for more accurate and efficient diagnoses by examining both high-level architectural patterns and cellular-level details.

The digital nature of WSIs has enabled a wide spectrum of computational pathology tasks [1, 8]. At the *patch level*, tasks such as cell nuclei segmentation [27, 31] and tissue classification [43] can be effectively addressed using standard computer vision models. At the slide level, tasks range from basic analysis like tumor detection, subtyping, and grading [4, 6, 33, 34] to more advanced applications such as biomarker prediction [10, 14, 23] and survival prediction [9, 25, 37]. While basic tasks generally have reliable labels corresponding directly to observable histological features, advanced tasks introduce additional challenges as they rely on clinical data or genomic profiling, creating inherent limitations in dataset construction and feature-outcome correlations.

However, analyzing WSIs at the slide level presents significant computational challenges. Due to their enormous size, often exceeding billions of pixels, direct analysis is computationally intensive and impractical with conventional computer vision methods designed for natural images. Moreover, large areas of WSIs may contain background or non-diagnostic information, necessitating efficient processing techniques that focus on informative regions. While transformer-based models [7, 12, 13] have shown remarkable capabilities in modeling long-range dependencies [11, 15, 29], their quadratic computational complexity makes direct application to gigapixel-scale WSIs impractical [42].

To address these challenges, Multiple Instance Learning (MIL) has emerged as a prevalent solution, with various approaches developed to process WSIs efficiently. The most straightforward approach divides WSIs into smaller patches and treats them as independent and identically distributed (i.i.d.) samples, processing them independently using multilayer perceptrons [8, 32] (Fig. 1, Top). While computationally efficient, this position-agnostic paradigm fails to capture the complex spatial relationships between patches, potentially losing critical diagnostic information.

Recent developments in WSI analysis primarily follow two directions. The first extends Attention-based MIL (ABMIL) through various architectural modifications, including instance-level supervision [32], dual-stream architectures [24], and advanced training strategies [40, 48, 49]. However, these methods remain constrained by treating patches as independent samples and fail to capture spatial relationships. The second direction attempts to preserve spatial context by reshaping the sparsely distributed patches into a large dense square, enabling the use of convolutional neural networks (CNNs) to aggregate local contextual information [38, 41] (Fig. 1, Middle). However, this artificial reshaping distorts the inherent spatial relationships between patches, as the original spatial distribution in WSIs is irregular.

Advances in transformer architectures have revolutionized the machine learning landscape across domains. Sparse attention variants [5, 30, 47] have demonstrated remarkable success by offering efficient computation while preserving the transformer's powerful modeling capabilities. Similarly, hierarchical modeling approaches have become dominant in both traditional CNN designs with progressive downsampling [16, 17, 26, 44] and modern transformer-based architectures [30, 45]. However, despite their success in general domains, these approaches face significant challenges when applied to WSIs, as they are designed for dense, uniformly distributed data, making them fundamentally ill-suited for WSIs where informative regions are sparsely and irregularly scattered across large backgrounds. Consequently, WSI analysis cannot benefit from these technical advances directly. Recent Transformer-based methods in WSI analysis attempt to bridge this gap by forcibly converting sparse inputs into dense squares, making it possible to leverage these advances from the general domain. For instance, TransMIL employs re-squaring with Nyström attention and CLS tokens, while RRT builds upon TransMIL by introducing region attention and cross-region attention mechanisms. Nevertheless, these approaches not only distort real positional relationships but also produce only isotropic representations, failing to take full advantage of hierarchical modeling capabilities that have proven crucial in general computer vision tasks.

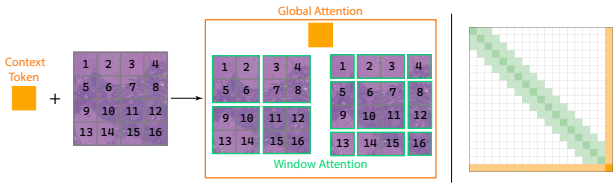

Figure 2. Schematic of sparse attention computation within our proposed framework. Left: The input WSI is partitioned into non-overlapping $2w \times 2w$ windows using an index-driven approach that leverages the inherent sparsity of WSIs. The windows are then shifted by $w \times w$ to obtain a second set of non-overlapping windows. Local attention is computed within each window (green boxes), while global attention is captured via a learnable global token (orange box) that interacts with all tokens in the WSI. Right: The attention pattern matrix visualizes the interaction scope of different tokens - the diagonal green blocks indicate that local tokens can only attend to other tokens within the windows, while the orange row& column show that the global token has unrestricted attention to all tokens in the WSI, enabling the capture of long-range dependencies.

Inspired by these advances and addressing the limitations of previous methods, we propose the Sparse Pyramid Attention Network (SPAN), a hierarchical approach that preserves the natural spatial distribution of patches while capturing multi-scale contextual information (Fig. 1, Bottom). SPAN comprises two key modules: Spatial-Adaptive Feature Condensation (SAC) and Context-Aware Feature Refinement (CAR). SAC progressively builds multi-scale feature representations through successive sparse downsampling operations on informative regions. Starting from single-scale input features, this process creates a feature hierarchy where deeper layers observe increasingly larger receptive fields, capturing both fine-grained cellular patterns and broader tissue architectures while significantly reducing the number of tokens for efficient subsequent processing. CAR then incorporates shifted windows and global context tokens to comprehensively refine these hierarchically condensed representations at each scale (Fig. 2). The shifted-window scheme captures dependencies beyond local regions, while global tokens enable cross-scale informa-

tion exchange, effectively modeling relationships that traditional patch-based methods often overlook. This two-stage design creates a natural synergy: SAC provides CAR with multi-scale features and manageable token counts, while CAR enriches these features with essential contextual information.

We evaluated SPAN on multiple public histopathology datasets [2–4, 6, 33, 34] across two basic visual tasks, classification and segmentation. Our experiments demonstrate that SPAN achieves superior performance compared to state-of-the-art methods. In summary, our contributions are as follows:

- We develop a novel rulebook-based computational framework for WSI analysis that preserves exact spatiality and can directly leverage advances from general domains, overcoming limitations of existing approaches that distort spatiality while enabling pyramid representation modeling previously inaccessible to WSI analysis.
- We propose SPAN, a cascaded feature learning architecture with SAC and CAR modules that progressively constructs multi-scale representations through spatial-adaptive condensation, enriches them with comprehensive contextual relationships, and efficiently directs computational resources to critical diagnostic regions.
- We demonstrate SPAN's superior performance over state-of-the-art methods through extensive evaluation on multiple public histopathology datasets across various computational pathology tasks.

2. Method

2.1. Overview

We now detail our rulebook based framework and SPAN's architecture. As illustrated in Fig. 3, SPAN processes WSIs through alternating SAC and CAR modules, with each iteration operating at a progressively coarser scale.

The SAC module serves a dual purpose: it not only performs spatial condensation through downsampling operations but also conducts coarse-grained feature transformation to capture basic visual patterns. This controlled dimension reduction maintains essential diagnostic information while providing initial feature representations. Following each condensation operation, the CAR module employs transformer blocks with shifted windows and a global token to perform fine-grained feature refinement, capturing subtle tissue patterns and their long-range dependencies at the current scale. This complementary design, where SAC handles both spatial condensation and coarse feature transformation while CAR focuses on detailed contextual refinement, enables efficient modeling of relationships between informative regions while avoiding redundant computations on uninformative areas.

This hierarchical processing repeats with subsequent SAC-CAR modules operating on increasingly condensed representations. The gradual reduction in spatial resolution, coupled with maintained feature dimensionality, allows SPAN to efficiently manage memory consumption while preserving multi-scale diagnostic patterns. The following sections detail the specific mechanisms within each module and their implementations.

2.2. Spatial-Adaptive Feature Condensation

The SAC module progressively condenses patches into more compact representations through learnable feature transformations. The design of SAC is motivated by two key insights: the inherent multi-scale nature of histopathological diagnosis that pathologists perform, and the computational efficiency required for processing large-scale WSIs. This motivates us to design an adaptive feature extraction process that can handle the irregular spatial distribution of tissue regions.

Our condensation process maintains spatial relationships while progressively reducing spatial dimensions to capture multi-scale patterns. To achieve this efficiently, we implement SAC using sparse convolutions [28] for downsampling and hierarchical feature encoding. This choice naturally aligns with the WSI structure, where significant background portions contain uninformative regions, enabling selective computation only where meaningful features are present.

2.2.1. Sparse Convolution Rulebook

Sparse convolution operations are typically implemented using a rulebook-based approach, which efficiently manages the computation and memory usage for sparse data structures. Specifically, an index matrix $\mathbf{I} = \begin{bmatrix}1 & 2 & \cdots & N\end{bmatrix}^{\mathrm{T}}$ corresponds to the coordinate matrix $\mathbf{P} = [p_i \mid i \in \mathbf{I}] \in \mathbb{N}^{N\times 2}$ and the feature matrix $\mathbf{X} = [x_i \mid i \in \mathbf{I}] \in \mathbb{R}^{N\times d}$. This structured representation ensures efficient access to coordinates and their associated features during sparse convolution operations.

For each convolutional layer, the output coordinates are computed based on the input coordinates, the kernel size K, the dilation D, and the layer's stride S:

$$\mathbf{P}_{\text{out}} = \{p_{i_{\text{out}}} \mid p_{i_{\text{out}}} = \left\lfloor \frac{p_{i_{\text{in}}} - (K-1)\cdot D}{S} \right\rfloor, \forall p_{i_{\text{in}}} \in \mathbf{P}_{\text{in}}\}, \tag{1}$$

where $\lfloor\cdot\rfloor$ denotes the floor operation, and $(K-1)\cdot D$ adjusts for the expansion of the receptive field due to the kernel size and dilation. The corresponding output indices $\mathbf{I}_{\text{out}}$ are assigned sequentially starting from 1.

To determine the valid mappings between input and output indices for each kernel offset, we construct a *rulebook* $\mathcal{R}_k$ defined as:

$$\mathcal{R}_k = \{(i_{\text{in}}, i_{\text{out}}) \mid p_{i_{\text{in}}} + k = p_{i_{\text{out}}}\}, \quad k \in \mathcal{K}, \tag{2}$$

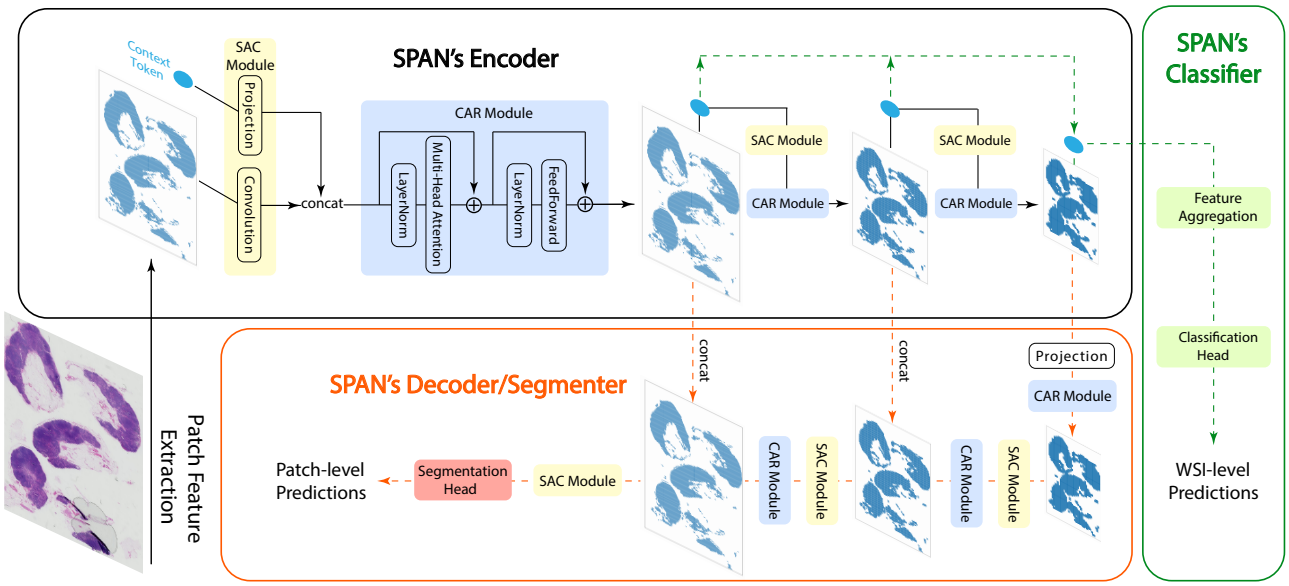

Figure 3. Overall architecture of SPAN. The encoder begins with a SAC (Spatial-Adaptive Feature Condensation) module comprising Projection and Convolution components, followed by CAR (Context-Aware Feature Refinement) that employs window attention through LayerNorm, Multi-Head Attention, and Feed-Forward layers for local context modeling. While the initial SAC preserves spatial dimensions with 1×1 convolution, subsequent SAC modules progressively downsample tokens to approximately 1/4 of their previous token count. This SAC-CAR sequence repeats multiple times for hierarchical feature extraction and refinement. Task-specific paths (dashed lines) enable flexible downstream applications: the decoder/segmenter path utilizes alternating CAR-SAC modules with transposed convolutions in SAC for upsampling and patch-level predictions, while the classifier path employs feature aggregation for WSI-level predictions.

where $\mathcal{K}$ is the set of kernel offsets, and $p_{i_{\text{in}}}$ and $p_{i_{\text{out}}}$ are input and output coordinates, respectively. Each entry in $\mathcal{R}_k$ represents an atomic operation, specifying that the input position $p_{i_{\text{in}}}$ shifted by the kernel offset k matches the output position $p_{i_{\text{out}}}$. The complete rulebook $\mathcal{R}_{\mathcal{K}} = \bigcup_{k \in \mathcal{K}} \mathcal{R}_k$ efficiently encodes the locations and conditions under which convolution operations are to be performed.

Each sparse convolutional layer performs convolution by executing the atomic operations defined in the rulebook $\mathcal{R}_{\mathcal{K}}$. An atomic operation $(i_{\text{in}}, i_{\text{out}}) \in \mathcal{R}_k$ transforms the input feature $h_{i_{\text{in}}}$ using the corresponding weight matrix $W_l(k)$ and accumulates the result to the output feature $h_{i_{\text{out}}}$. The complete sparse convolution operation for a layer l is defined as:

$$h_{i_{\text{out}}} = \sum_{k \in \mathcal{K}} \sum_{\mathcal{R}_k} W_l(k) h_{i_{\text{in}}} + b_l, \tag{3}$$

where $h_{i_{\text{in}}} \in \mathbb{R}^{d_{\text{in}}}$ is the input feature at index i_{in}, $h_{i_{\text{out}}} \in \mathbb{R}^{d_{\text{out}}}$ is the output feature at index i_{out}, $W_l(k) \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the weight matrix associated with kernel offset k, and $b_l \in \mathbb{R}^{d_{\text{out}}}$ is the bias term for layer l.

By utilizing this rulebook-based approach, the sparse convolutional layer efficiently aggregates information from neighboring input features by performing computations only at the necessary locations. This method effectively captures local spatial patterns in the sparse data while significantly reducing computational overhead and memory usage compared to dense convolution operations, as it avoids unnecessary calculations in empty or uninformative regions. For the context token, we compute and average features with all kernel weights and biases if dimension reduction is needed. Otherwise, we maintain an identity projection.

2.3. Context-Aware Feature Refinement

The Context-Aware Feature Refinement (CAR) module builds upon the condensed feature representation to model comprehensive contextual relationships. While the preceding SAC module efficiently captures hierarchical features through progressive condensation, the refined understanding of histological patterns requires modeling both local tissue structures and their long-range dependencies. This dual modeling requirement motivates us to adopt attention mechanisms, which excel at capturing both local and long-range dependencies through learnable interactions between features.

To effectively implement the CAR module, we face several technical challenges in applying attention mechanisms to WSI analysis. Traditional sparse attention approaches [5, 30, 47], despite their success in various do-

mains, operate on dense feature matrices by striding over fixed elements in the matrix's memory layout. This approach would require densifying our sparse WSI features and applying padding operations to match the fixed memory layout. Given the high feature dimensionality characteristic of WSI analysis, such transformation would introduce substantial memory and computational overhead while compromising the efficiency established in the previous SAC module.

Therefore, we develop a sparse attention rulebook that directly operates on the sparse feature representation, maintaining compatibility with the SAC module's index-coordinate system. Our approach leverages $\mathbf{I}$ and $\mathbf{P}$ inherited from previous layers to define sparse attention windows, where features within each window can attend to each other without dense transformations. This design preserves both computational efficiency and the sparse structure compatibility established in earlier modules.

2.3.1. Sparse Attention Rulebook

To efficiently handle sparse data representations, we formulate attention computation using rulebooks following the paradigm of sparse convolutions. The first step is to generate attention windows that define which tokens should attend to each other. For efficient window generation, we temporarily densify $\mathbf{I} \in \mathbb{N}^N$ into a regular grid using patch coordinates $\mathbf{P} \in \mathbb{N}^{N\times 2}$ with zero padding. This enables efficient block-wise memory access on a low-dimensional index matrix rather than operating on a high-dimensional feature matrix. As illustrated in Figure 2, we stride over the densified index matrix to generate regular and shifted windows, where the shifting operation ensures comprehensive coverage of local contexts. The resulting $\mathcal{W}$ is a collection of windows, where each window contains a set of patch indices excluding padded zeros. These windows effectively define the grouping of indices for constructing an attention rulebook.

To enhance the model's ability to capture global dependencies, we introduce a learnable global context token that provides a shared context accessible to all other tokens. The combined hidden features can be represented as $\mathbf{H} = [h_{i_1}^\top, h_{i_2}^\top, \ldots, h_{i_N}^\top, h_g^\top] \in \mathbb{R}^{(N+1)\times d_{\text{out}}}$, where h_g denotes the global context token. For self-attention computation, we project $\mathbf{H} \in \mathbb{R}^{(N+1)\times d}$ into $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ using linear projections.

Having defined the attention windows, we now construct two types of rulebooks to capture both local and global dependencies. For local attention, the rulebook $\mathcal{R}_w$ for each window is defined as:

$$\mathcal{R}_w = \{(i,j) \mid i,j \in w\}, \quad w \in \mathcal{W}, \tag{4}$$

where $\mathcal{W}$ denotes the set of all attention windows, and i and j represent the indices of the input and output patches within the window w, respectively. Each entry $(i,j) \in \mathcal{R}_w$ represents a local attention atomic operation between tokens i and j. These atomic operations are defined by the following equations. The attention scores are computed with a learnable relative positional bias to account for spatial relationships:

$$e_{ij}^{\text{local}} = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}} + B(p_i - p_j), \tag{5}$$

where $\mathbf{q}_i$ and $\mathbf{k}_j$ represent the query and key vectors for local tokens i and j, respectively, and p_i and p_j denote their positions. $B(p_i - p_j)$ represents the learnable relative positional biases (RPB) [30], parameterized by a matrix $B \in \mathbb{R}^{(2w_{size}-1)\times(2w_{size}-1)\times \text{num\_heads}}$.

The choice of positional encoding is crucial for capturing spatial relationships in WSI analysis. RPB enhances the model's ability to recognize positional nuances and disrupt the permutation invariance inherent in self-attention mechanisms while maintaining parameter efficiency. Alternative approaches present different trade-offs: absolute positional encoding (APE) [13] would significantly increase the parameter count given the extensive spatial dimension of possible positions in WSIs, while Rotary Position Embedding (RoPE) [18, 39] and Attention with Linear Biases (Alibi) [35], despite their parameter efficiency in language models, prove less effective at capturing spatial relationships in our context.

The final output of the local attention is then computed as:

$$\mathbf{h}_i^{\text{local}} = \sum_{w\in\mathcal{W}} \sum_{j:(i,j)\in\mathcal{R}_w} \frac{\exp(e_{ij}^{\text{local}})}{\sum_{k:(i,k)\in\mathcal{R}_{\text{local}}} \exp(e_{ik}^{\text{local}})} \mathbf{v}_j. \tag{6}$$

To complement local attention with global context modeling, we introduce global attention that operates on all patch tokens and the learnable global context token. The global attention rulebook is defined as:

$$\mathcal{R}_g = \{(i,j),(j,i) \mid i \in [1,N], j \in \{N+1\}\}, \tag{7}$$

The global attention mechanism employs similar formulations as equations (5) and (6) but excludes the positional bias term, yielding $\mathbf{h}_i^{global}$. While local attention is constrained to windows, global attention spans across the entire feature map through the global context token, enabling comprehensive contextual integration. The final output features combine both local and global dependencies through:

$$\mathbf{h}_i^{\text{out}} = \mathbf{h}_i^{local} + \mathbf{h}_i^{global}. \tag{8}$$

2.4. Integration with Downstream Tasks

SPAN's hierarchical feature representation readily adapts to both classification and segmentation tasks. For classification, our SAC modules capture multi-scale features from

cellular details to tissue architectures. We obtain slide-level representations by summing context tokens across different depths, effectively integrating information at all granularities while maintaining simplicity. This approach achieves strong performance while remaining compatible with more sophisticated training strategies like those in ABMIL. For segmentation, SPAN naturally extends to a U-Net [36] architecture through its hierarchical sparse design. The decoder path uses sparse transpose convolutions to recover spatial resolution efficiently, while skip connections combine fine spatial details with semantic context from different stages. This design enables accurate tissue region delineation while maintaining the computational advantages of our sparse approach.

3. Experiments

3.1. Tasks and Datasets

We evaluate SPAN on both classification and segmentation tasks using multiple public WSI datasets. For classification, we address three diagnostic tasks: tumor detection (CAMELYON16 [4]), tumor grading (BRACS [6]), and tumor subtyping (TCGA-Lung [33, 34]). For segmentation, we use CAMELYON16, CAMELYON17 [3], and BACH [2] datasets. From CAMELYON16/17, we exclude tumor-positive slides lacking pixel-level annotations to ensure reliable evaluation. We additionally curate SegCAMELYON, containing only fully annotated tumor-positive slides from both CAMELYON datasets, to specifically evaluate tumor boundary delineation capabilities.

3.2. Experimental Setup

Our experimental protocol extends standard WSI analysis practices with key improvements for reproducibility and clinical relevance. The preprocessing pipeline builds upon CLAM's [32] approach, adding a grid alignment step that extends patch boundaries to the nearest multiple of 224 pixels. This ensures precise spatial coordinates essential for our grid-based representation. For patch-level ground truth generation, patches with tumor tissue exceeding 20% area are labeled as positive. For feature extraction, we use a pretrained ResNet50 encoder. We implement 3-layer GCN and GAT models with 8-adjacent connectivity as additional baselines for segmentation, following standard practices in WSI analysis [9, 19, 46]. To ensure robust evaluation, we conduct experiments with random initialization and dataset splits. We combine all available data and perform random stratified splits. Each experiment is repeated with 5 different random seeds controlling both model initialization and data splitting. Model selection is determined using validation set performance, with predictions made through direct class probability argmax without post-hoc threshold optimization, to mirror real-world deployment scenarios.

3.3. Main Results

Tables 1 and 2 present comprehensive evaluations across classification tasks. Methods are organized by their foundational architectures, with ABMIL representing the classical attention-based MIL approach and its variants. While ABMIL-based methods rely on auxiliary losses and sophisticated training strategies, SPAN achieves strong performance with simpler training objectives, demonstrating the effectiveness of our architectural design. Notably, our spatial-aware architecture shows substantial improvements on the challenging multi-class BRACS dataset, highlighting the importance of effective spatial modeling in complex diagnostic scenarios. For segmentation tasks (Table 3), SPAN demonstrates superior performance across all datasets in both Dice and IoU metrics. This success stems from our undistorted spatial encoding, which preserves precise patch relationships crucial for segmentation boundaries, and the hierarchical design that enables effective multi-scale contextual modeling. These architectural advantages particularly benefit tasks requiring precise spatial localization.

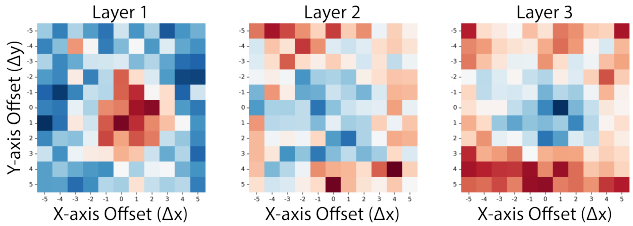

Figure 4. Layer-wise visualization of learned RPB in SPAN. Each heatmap shows the attention bias values as a function of relative positional offsets (Δx, Δy) between token pairs, where coordinates (x, y) represent the bias when attending to a token at x positions horizontally and y positions vertically relative to the query token. Red and blue colors indicate higher and lower attention biases, respectively.

To understand how our model captures spatial dependencies, we visualize the learned relative position bias (RPB) patterns across transformer layers (Fig. 4). The analysis reveals a systematic progression: lower layers exhibit concentrated attention around (Δx, Δy) = (0, 0), focusing on local spatial relationships, while deeper layers show stronger biases in peripheral regions. This evolution from local to global attention patterns enables the model to effectively process both fine-grained cellular details and broader tissue architectures. Unlike fixed positional encoding schemes, this data-driven approach allows more flexible adaptation to the hierarchical nature of histopathological images.

3.4. Ablation Studies

To assess the contributions of various components, we conducted ablation studies using the CAMELYON16 dataset with a fixed window size of 6×6. We examine different

Table 1. Classification performance on CAMELYON16 and TCGA-Lung datasets

Method	CAMELYON16			TCGA-Lung		
	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score
ABMIL backbone						
ABMIL[21]	0.857 ± 0.085	0.915 ± 0.059	0.850 ± 0.088	0.879 ± 0.024	0.947 ± 0.018	0.878 ± 0.025
CLAM-SB[32]	0.873 ± 0.040	0.922 ± 0.058	0.868 ± 0.039	0.874 ± 0.009	0.948 ± 0.016	0.873 ± 0.010
CLAM-MB[32]	0.867 ± 0.031	0.932 ± 0.023	0.862 ± 0.031	0.879 ± 0.025	0.953 ± 0.014	0.877 ± 0.025
DTFD [48]	0.877 ± 0.073	0.947 ± 0.039	0.868 ± 0.057	0.834 ± 0.039	0.927 ± 0.025	0.830 ± 0.042
DSMIL[24]	0.887 ± 0.051	0.941 ± 0.025	0.881 ± 0.050	0.884 ± 0.018	0.951 ± 0.022	0.883 ± 0.018
MHIM[40]	0.883 ± 0.053	0.929 ± 0.036	0.877 ± 0.056	0.884 ± 0.021	0.945 ± 0.014	0.883 ± 0.022
ACMIL[49]	0.893 ± 0.015	0.936 ± 0.023	0.889 ± 0.011	0.879 ± 0.020	0.947 ± 0.010	0.877 ± 0.020
GNN backbone						
PatchGCN[9]	0.833 ± 0.065	0.874 ± 0.076	0.819 ± 0.072	0.870 ± 0.015	0.942 ± 0.012	0.869 ± 0.015
TransMIL backbone						
TransMIL[38]	0.873 ± 0.053	0.916 ± 0.056	0.867 ± 0.053	0.863 ± 0.010	0.932 ± 0.006	0.863 ± 0.011
RRT[41]	0.867 ± 0.029	0.936 ± 0.038	0.862 ± 0.027	0.879 ± 0.044	0.953 ± 0.018	0.879 ± 0.044
SPAN backbone						
SPAN	0.903 ± 0.030	0.939 ± 0.026	0.898 ± 0.032	0.886 ± 0.025	0.950 ± 0.017	0.885 ± 0.025

Table 2. Classification performance on the BRACS dataset

Method	BRACS					
	Accuracy	AUC (Negative)	AUC (Atypical)	AUC (Positive)	Macro AUC	Macro F1
ABMIL backbone						
ABMIL	0.687 ± 0.023	0.856 ± 0.038	0.718 ± 0.031	0.910 ± 0.028	0.828 ± 0.099	0.552 ± 0.039
CLAM-SB	0.687 ± 0.044	0.877 ± 0.031	0.728 ± 0.065	0.916 ± 0.025	0.840 ± 0.099	0.562 ± 0.041
CLAM-MB	0.696 ± 0.039	0.877 ± 0.027	0.752 ± 0.045	0.913 ± 0.031	0.847 ± 0.085	0.545 ± 0.049
DTFD	0.689 ± 0.027	0.863 ± 0.020	0.698 ± 0.029	0.922 ± 0.021	0.828 ± 0.116	0.578 ± 0.034
DSMIL	0.699 ± 0.035	0.859 ± 0.022	0.713 ± 0.056	0.908 ± 0.026	0.826 ± 0.101	0.553 ± 0.056
MHIM	0.716 ± 0.028	0.884 ± 0.010	0.731 ± 0.041	0.927 ± 0.020	0.847 ± 0.103	0.560 ± 0.066
ACMIL	0.720 ± 0.022	0.881 ± 0.033	0.765 ± 0.024	0.931 ± 0.018	0.859 ± 0.085	0.604 ± 0.074
GNN backbone						
PatchGCN	0.713 ± 0.025	0.882 ± 0.016	0.735 ± 0.035	0.928 ± 0.019	0.848 ± 0.101	0.610 ± 0.031
TransMIL backbone						
TransMIL	0.692 ± 0.037	0.846 ± 0.085	0.665 ± 0.060	0.885 ± 0.058	0.799 ± 0.117	0.577 ± 0.034
RRT	0.718 ± 0.036	0.880 ± 0.029	0.744 ± 0.064	0.922 ± 0.023	0.848 ± 0.093	0.595 ± 0.065
SPAN backbone						
SPAN	0.740 ± 0.052	0.894 ± 0.023	0.766 ± 0.035	0.926 ± 0.023	0.862 ± 0.085	0.644 ± 0.082

positional encoding techniques, the downsampling convolutions, and the shifted-window mechanism, demonstrating our framework's compatibility with general advances. Through systematic modification of these components, we explored their impact on model performance.

The ablation results (Table 4) reveal several key insights. First, while positional encoding enhances performance, our model remains competitive even without it, attributable to the inherent spatial information captured by the window attention and convolutions. Among all positional encoding methods, RPB achieves the best performance. The inferior performance of Axial RoPE and Alibi might stem from two main limitations. Their predefined frequencies, inherited from NLP domains, do not optimally align with the spatial characteristics of WSIs. Furthermore, their distance-based attention decay mechanism contradicts our observed RPB

Table 3. Segmentation performance on histopathology datasets

Method	CAMELYON16		CAMELYON17		SegCAMELYON		BACH	
	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU
ABMIL†	0.742±0.012	0.591±0.016	0.548±0.136	0.387±0.120	0.738±0.038	0.586±0.047	0.690±0.158	0.544±0.181
TransMIL†	0.822±0.051	0.700±0.071	0.754±0.133	0.618±0.156	0.818±0.055	0.695±0.079	0.723±0.176	0.588±0.201
RRT†	0.836±0.062	0.722±0.094	0.786±0.118	0.660±0.154	0.829±0.066	0.712±0.100	0.705±0.128	0.557±0.159
GCN	0.841±0.006	0.726±0.010	0.754±0.080	0.610±0.103	0.809±0.068	0.684±0.098	0.695±0.169	0.552±0.191
GAT	0.795±0.029	0.661±0.040	0.838±0.058	0.724±0.087	0.805±0.045	0.676±0.064	0.715±0.136	0.571±0.168
SPAN	0.885±0.043	0.796±0.069	0.870±0.038	0.771±0.061	0.860±0.052	0.757±0.080	0.783±0.137	0.659±0.173

† Method name indicates its corresponding architecture: ABMIL for MLP, TransMIL for vanilla Nystromformer, and RRT for region-based Nystromformer.

Table 4. Ablation study results for different model configurations

Configuration	Accuracy	AUC
Positional Encoding		
Axial Alibi	0.883 ± 0.039	0.920 ± 0.029
Axial RoPE	0.880 ± 0.048	0.917 ± 0.017
None	0.890 ± 0.019	0.938 ± 0.027
SAC Module		
No Downsampling	0.879 ± 0.037	0.928 ± 0.026
CAR Module		
No Shifted Window	0.883 ± 0.039	0.923 ± 0.049

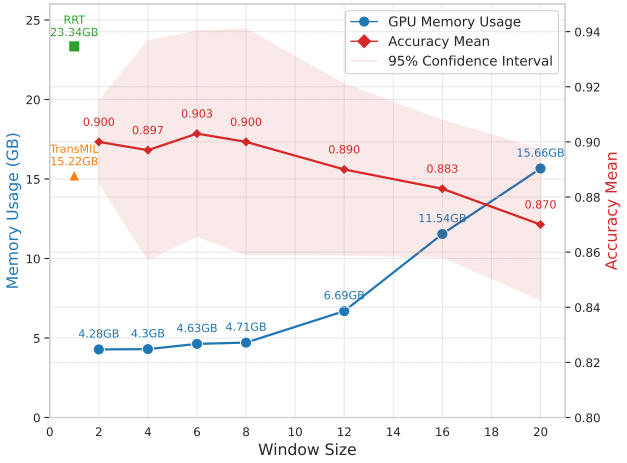

Figure 5. Accuracy and memory usage of SPAN with window sizes from 2×2 to 20×20. Each configuration is evaluated over 5 runs, with the mean accuracy and peak memory usage reported.

patterns in Fig. 4, where long-distance attention systematically strengthens in deeper layers. Such dynamic behavior cannot be achieved by fixed positional encodings.

Additionally, both the shifted-window mechanism and pyramid structure prove crucial, with their removal leading to notable performance degradation. Our findings demonstrate that, similar to general computer vision domains, pyramid representations are superior to isotropic ones for WSI analysis, enabling effective multi-scale feature representation where different tissue features manifest at varying magnification levels. Likewise, cross-window communication proves essential for WSI analysis where adjacent tissue regions exhibit strong diagnostic correlations.

We further investigate the impact of window size configurations on model performance and efficiency. As shown in Fig. 5, increasing the window size beyond a certain point does not necessarily improve performance in our settings; however, it significantly increases memory usage. This may be attributed to insufficient training data to learn complex feature interactions effectively at larger window sizes.

4. Discussion and Conclusion

Our experiments validate that fundamental principles successful in general computer vision can be effectively adapted for WSI analysis. While previous WSI frameworks could only approximate these techniques, our proposed rulebook-based framework enables direct application of hierarchical modeling and attention mechanisms while preserving the original spatial relationships. This capability to exactly leverage rather than loosely mimic general deep learning advances demonstrates significant potential for improving computational pathology systems and establishes a stronger bridge between these domains previously hindered by the unique natures of gigapixel WSIs.

Building upon these architectural contributions, future research could explore additional technical refinements, such as WSI-specific adaptations to positional encoding methods derived from NLP, including learnable or carefully designed frequency patterns that better match the hierarchical nature of histological images. Beyond technical improvements, integrating additional clinical and molecular data could enable broader applications beyond pure visual analysis. We hope this work will inspire further research into architectures for WSI analysis, ultimately contributing to computational pathology solutions that leverage both visual and non-visual information.

References

[1] Esther Abels, Liron Pantanowitz, Famke Aeffner, Mark D Zarella, Jeroen van der Laak, Marilyn M Bui, Venkata NP Vemuri, Anil V Parwani, Jeff Gibbs, Emmanuel Agosto-Arroyo, et al. Computational pathology definitions, best practices, and recommendations for regulatory guidance: a white paper from the digital pathology association. *The Journal of Pathology*, 2019. 1

[2] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, et al. Bach: Grand challenge on breast cancer histology images. *Medical Image Analysis*, 2019. 3, 6

[3] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *Transactions on Medical Imaging*, 2018. 6

[4] 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*, 2017. 1, 3, 6

[5] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 2, 4

[6] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, et al. Bracs: A dataset for breast carcinoma subtyping in h&e histology images. *Database*, 2022. 1, 3, 6

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. 2

[8] 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*, 2019. 1, 2

[9] Richard J Chen, Ming Y Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks. In *Medical Image Computing and Computer Assisted Intervention*, 2021. 1, 6, 7

[10] Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L Moreira, Narges Razavian, and Aristotelis Tsirigos. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature Medicine*, 2018. 1

[11] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024. 2

[12] 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. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 5

[14] Omar SM El Nahhas, Marko van Treeck, Georg Wölflein, Michaela Unger, Marta Ligero, Tim Lenz, Sophia J Wagner, Katherine J Hewitt, Firas Khader, Sebastian Foersch, et al. From whole-slide image to biomarker prediction: end-to-end weakly supervised deep learning in computational pathology. *Nature Protocols*, 2024. 1

[15] Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. Fastervit: Fast vision transformers with hierarchical attention. In *International Conference on Learning Representations*, 2024. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 2015. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2

[18] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. *arXiv preprint arXiv:2403.13298*, 2024. 5

[19] Wentai Hou, Lequan Yu, Chengxuan Lin, Helong Huang, Rongshan Yu, Jing Qin, and Liansheng Wang. Hˆ 2-mil: exploring hierarchical representation with heterogeneous multiple instance learning for whole slide image analysis. In *AAAI Conference on Artificial Intelligence*, 2022. 6

[20] 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. 3

[21] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, 2018. 7

[22] Md Amirul Islam*, Sen Jia*, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? In *International Conference on Learning Representations*, 2020. 2

[23] Darui Jin, Shangying Liang, Artem Shmatko, Alexander Arnold, David Horst, Thomas GP Grünewald, Moritz Gerstung, and Xiangzhi Bai. Teacher-student collaborated multiple instance learning for pan-cancer pdl1 expression prediction from histopathology slides. *Nature Communications*, 2024. 1

[24] 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 *Conference on Computer Vision and Pattern Recognition*, 2021. 2, 7

[25] Zhe Li, Yuming Jiang, Mengkang Lu, Ruijiang Li, and Yong Xia. Survival prediction via hierarchical multimodal co-attention transformer: A computational histology-radiology solution. *Transactions on Medical Imaging*, 2023. 1

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Conference on Computer Vision and Pattern Recognition*, 2017. 2

[27] Yi Lin, Zeyu Wang, Dong Zhang, Kwang-Ting Cheng, and Hao Chen. Bonus: Boundary mining for nuclei segmentation with partial point labels. *Transactions on Medical Imaging*, 2024. 1

[28] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*, 2015. 3

[29] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 2

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision*, 2021. 2, 4, 5

[31] Wei Lou, Xiang Wan, Guanbin Li, Xiaoying Lou, Chenghang Li, Feng Gao, and Haofeng Li. Structure embedded nucleus classification for histopathology images. *Transactions on Medical Imaging*, 2024. 1

[32] 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*, 2021. 2, 6, 7

[33] Cancer Genome Atlas Research Network et al. Comprehensive genomic characterization of squamous cell lung cancers. *Nature*, 2012. 1, 3, 6

[34] Cancer Genome Atlas Research Network et al. Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, 2014. 1, 3, 6

[35] Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. 5

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer Assisted Intervention*, 2015. 6

[37] Wei Shao, YangYang Shi, Daoqiang Zhang, JunJie Zhou, and Peng Wan. Tumor micro-environment interactions guided graph learning for survival analysis of human cancers from whole-slide pathological images. In *Conference on Computer Vision and Pattern Recognition*, 2024. 1

[38] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. In *Advances in Neural Information Processing Systems*, 2021. 2, 7

[39] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 5

[40] 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 *International Conference on Computer Vision*, 2023. 2, 7

[41] 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 *Conference on Computer Vision and Pattern Recognition*, 2024. 2, 7

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2

[43] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention*, 2018. 1

[44] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[45] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *International Conference on Computer Vision*, 2021. 2

[46] Weiyi Wu, Xiaoying Liu, Robert B Hamilton, Arief A Suriawinata, and Saeed Hassanpour. Graph convolutional neural networks for histologic classification of pancreatic cancer. *Archives of Pathology & Laboratory Medicine*, 2023. 6

[47] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, 2020. 2, 4

[48] 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. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2, 7

[49] 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. 2, 7

Enhancing Digital Pathology Visual Understanding With Sparse Pyramid Attention Networks

Supplementary Material

5. Implementation Details

5.1. Layer Configurations

In the first block, we apply a channel-wise 1×1 sparse convolution to the input features, which maintains the original spatial resolution for preserving fine-grained details while reducing the feature dimension. For the subsequent blocks, we utilize sparse convolutions with kernel size $K = 2$, dilation rate $D = 2$, and stride $S = 2$ to progressively downsample the spatial dimensions. The dilated convolutions effectively expand the receptive field, enabling the network to capture broader contextual information during initial feature transformation. We maintain a constant feature dimension across these subsequent layers to preserve the depth of information. This configuration progressively forms coarse-grained hierarchical representations for subsequent refinement stages while maintaining computational efficiency.

5.2. Training Configuration

All models were trained for 50 epochs using an initial learning rate of 1e-4 and weight decay of 1e-5 for both classification and segmentation tasks. The hyperparameter settings are kept the same across different datasets. For model selection, we employed a validation-based strategy where the checkpoint achieving the highest accuracy was selected. In cases of equal accuracy, the checkpoint with the highest AUC score was chosen for final evaluation on the test set. To ensure robust evaluation, we conducted 5 independent runs with different random initializations. For each run, all samples were concatenated and then randomly split using seeds 0 through 4, which controlled both dataset partitioning and model parameter initialization. Our code will be made publicly available to facilitate reproducibility and future research.

5.3. Computational Analysis

Given that our method incorporates operations whose computational efficiency cannot be directly quantified through FLOPs analysis, we instead present runtime comparisons under identical experimental settings. On the Camelyon16 dataset (0.7 training, 0.15 evaluation split), our method achieves a runtime of 16.4 seconds per epoch, compared to TransMIL's 19.3 seconds. This demonstrates that, even with additional non-FLOPs operations, our approach remains competitive with existing architectures in terms of efficiency.

Moreover, our current implementation has room for further optimization. For instance, we compute sparse attention rulebooks online during each iteration. These computations could be precomputed and cached prior to training to enhance efficiency. Such optimizations, along with others, could further improve computational performance, when efficiency is a primary concern.

5.4. Classification Head Design

For the classification task, we aggregate information from the context tokens across L layers. Let $\mathbf{h}_l^g \in \mathbb{R}^d$ denote the global context token from layer l, where $l \in \{1, \ldots, L\}$. The final slide-level representation $\mathbf{h}^{\text{cls}}$ is computed as:

$$\mathbf{h}^{\text{cls}} = \sum_{l=1}^{L} \mathbf{h}_l^g \tag{9}$$

The classification prediction $\hat{y}$ is then obtained through a linear transformation:

$$\hat{y} = \text{softmax}(W^{\text{cls}}\mathbf{h}^{\text{cls}} + b^{\text{cls}}) \tag{10}$$

where $W^{\text{cls}} \in \mathbb{R}^{c \times d}$ and $b^{\text{cls}} \in \mathbb{R}^c$ are learnable parameters, and c is the number of classes.

5.5. Segmentation Head Design

As illustrated in Fig. 3 and briefly introduced in the methods section, our segmentation head employs a decoder that mirrors the encoder architecture to recover the original spatial resolution and generate patch-level predictions. The decoder follows a U-Net-like architecture with multi-scale feature integration.

In the decoder path, the primary distinction from the encoder lies in the SAC module, where sparse convolution is replaced by sparse deconvolution for upsampling. The CAR module remains identical to its encoder counterpart, providing context-aware refinement of the upsampled features.

Let $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_L$ denote the multi-scale feature representations extracted from the encoder, where $\mathbf{E}_l \in \mathbb{R}^{N_l \times d}$ represents the feature map at the l-th encoding level with N_l tokens. Here, $l = 1$ corresponds to the shallowest layer (highest spatial resolution), and $l = L$ corresponds to the deepest layer (lowest spatial resolution).

The decoder generates a sequence of feature representations $\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_L$, where $\mathbf{D}_l \in \mathbb{R}^{N_l \times d}$ corresponds to the output of the l-th decoding stage. The decoding process proceeds from low to high resolution, aligning with the encoder's multi-scale hierarchy.

At each decoding stage l, the input features $\mathbf{X}_l$ are processed through a SAC-CAR block:

$$\mathbf{D}_l = \text{SAC}(\text{CAR}(\mathbf{X}_l)) \in \mathbb{R}^{N_l \times d}. \tag{11}$$

For the first decoding stage ($l = 1$), the input features are obtained directly from the deepest encoder layer:

$$\mathbf{X}_1 = \mathbf{E}_L. \tag{12}$$

For subsequent decoding stages ($l > 1$), the input features $\mathbf{X}_l$ are formed by concatenating the upsampled decoder features from the previous stage with the corresponding encoder features:

$$\mathbf{X}_l = \mathbf{D}_{l-1} \parallel \mathbf{E}_{L-l+1} \in \mathbb{R}^{N_l \times 2d}, \tag{13}$$

where $\parallel$ denotes concatenation along the feature dimension. The SAC module then reduces the concatenated feature dimension from $2d$ back to d before feeding it into the CAR module.

The final segmentation prediction at each spatial location i is computed using the features from the highest resolution decoding stage ($l = L$):

$$\hat{y}_i = \text{softmax}(W^{\text{seg}} \mathbf{D}_L[i] + b^{\text{seg}}), \tag{14}$$

where $W^{\text{seg}} \in \mathbb{R}^{s \times d}$ and $b^{\text{seg}} \in \mathbb{R}^s$ are learnable parameters, and s is the number of segmentation classes.

In this formulation, $\mathbf{D}_l$ represents the hidden feature representations at the l-th decoding stage. By indexing both encoder and decoder stages from 1 to L, with increasing l corresponding to higher spatial resolutions, we provide a consistent and intuitive framework that aligns with common practices in the literature. This organization facilitates understanding of the multi-scale feature integration and the flow of information through the network.

5.6. Sparse Padding Implementation

In whole slide image (WSI) analysis, input patches often exhibit varying sizes due to the heterogeneity of tissue regions and sampling strategies. This variability poses challenges when applying convolutional operations across multiple layers, as certain patches may not be fully covered by the convolutional kernels, leading to ignored or misaligned features. This issue becomes more pronounced during the upsampling process in the segmentation path, potentially causing misalignment in the reconstructed feature maps.

Furthermore, prior research has demonstrated that padding can aid convolutional neural networks (CNNs) in learning positional information effectively [22], enabling models to differentiate between edges or corners and central regions of images. To preserve this advantageous property, we introduce a sparse padding strategy tailored for sparse data representations.

Our proposed sparse padding method is designed to maintain consistent spatial dimensions across convolutional layers while minimally increasing computational overhead. Specifically, in the context of sparse data, padding can be efficiently implemented by adding only a few additional points to the set of spatial coordinates. This approach ensures that the convolutional kernels can fully cover the input data at each layer, maintaining alignment and spatial consistency.

Algorithm 1: Sparse Padding Procedure

Input: Sparse coordinate set $\mathbf{P} \subset \mathbb{Z}^2$, feature set $\mathbf{F}$, kernel size k, stride s, dilation d, number of layers L

Output: Padded coordinate set $\mathbf{P}'$, padded feature set $\mathbf{F}'$

/* Compute Total Stride and Effective Receptive Field */

$S \leftarrow s^L$;

$R \leftarrow S \times (k-1) + 1$;

/* Determine Spatial Dimensions of Input */

$w \leftarrow \max_{(x,y)\in\mathbf{P}} x + 1$;

$h \leftarrow \max_{(x,y)\in\mathbf{P}} y + 1$;

/* Compute Necessary Padding Amounts for width and height */

$p_w \leftarrow (R - (w-1) \mod S) \mod S$;

$p_h \leftarrow (R - (h-1) \mod S) \mod S$;

/* Calculate Padding Offsets */

pad_left $\leftarrow \lfloor \frac{p_w}{2} \rfloor$;

pad_right $\leftarrow p_w -$ pad_left;

pad_top $\leftarrow \lfloor \frac{p_h}{2} \rfloor$;

pad_bottom $\leftarrow p_h -$ pad_top;

/* Generate padding coordinates */

$\mathbf{P}_{\text{pad}} \leftarrow \{(-\text{pad\_left}, -\text{pad\_top}), (w-1+\text{pad\_right}, h-1+\text{pad\_bottom})\}$;

/* Concatenate padding coordinates and original coordinates */

$\mathbf{P}' \leftarrow \mathbf{P} \cup \mathbf{P}_{\text{pad}}$;

/* Create corresponding padding features and concatenate */

$\mathbf{F}_{\text{pad}} = \mathbf{0}_{\dim(\mathbf{F})}$;

$\mathbf{F}' \leftarrow \mathbf{F} \cup \mathbf{F}_{\text{pad}}$;

/* Shift all coordinates to start from (0, 0) */

$\mathbf{P}' \leftarrow \mathbf{P}' + (\text{pad\_left}, \text{pad\_top})$;

This padding approach ensures that the convolutional operations can be applied without loss of information or misalignment, even when input sizes vary. By maintaining consistent spatial shapes across layers, we prevent the accumulation of spatial discrepancies that could adversely affect the segmentation quality. Moreover, the sparse nature

Table 5. Classification performance on CAMELYON16 and TCGA-Lung with PLIP as the feature extractor

Method	CAMELYON16			TCGA-Lung		
	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score
ABMIL backbone						
ABMIL	0.917 ± 0.020	0.957 ± 0.024	0.913 ± 0.022	0.881 ± 0.026	0.949 ± 0.022	0.880 ± 0.026
CLAM-SB	0.913 ± 0.038	0.949 ± 0.030	0.911 ± 0.037	0.883 ± 0.018	0.949 ± 0.016	0.881 ± 0.018
CLAM-MB	0.913 ± 0.032	0.959 ± 0.025	0.910 ± 0.032	0.881 ± 0.018	0.953 ± 0.012	0.880 ± 0.019
DTFD	0.917 ± 0.029	0.968 ± 0.026	0.910 ± 0.031	0.891 ± 0.021	0.966 ± 0.021	0.890 ± 0.021
DSMIL	0.923 ± 0.009	0.965 ± 0.025	0.920 ± 0.009	0.884 ± 0.017	0.959 ± 0.012	0.882 ± 0.020
MHIM	0.927 ± 0.025	0.969 ± 0.020	0.924 ± 0.024	0.892 ± 0.015	0.957 ± 0.009	0.891 ± 0.016
ACMIL	0.917 ± 0.041	0.959 ± 0.019	0.914 ± 0.040	0.888 ± 0.011	0.954 ± 0.005	0.887 ± 0.012
GNN backbone						
PatchGCN	0.913 ± 0.043	0.952 ± 0.029	0.909 ± 0.044	0.881 ± 0.035	0.948 ± 0.018	0.880 ± 0.036
TransMIL backbone						
TransMIL	0.907 ± 0.019	0.950 ± 0.025	0.903 ± 0.022	0.884 ± 0.016	0.947 ± 0.018	0.883 ± 0.017
RRT	0.920 ± 0.032	0.957 ± 0.019	0.917 ± 0.031	0.881 ± 0.021	0.951 ± 0.014	0.880 ± 0.022
SPAN backbone						
SPAN	0.937 ± 0.034	0.957 ± 0.025	0.934 ± 0.034	0.907 ± 0.016	0.964 ± 0.004	0.906 ± 0.017

of the padding minimizes the additional computational burden, making it suitable for large-scale WSI analysis where efficiency is crucial.

Our method effectively balances the need for spatial consistency and computational efficiency, providing a practical solution for handling variable-sized inputs in sparse convolutional networks. This strategy can be seamlessly integrated into existing frameworks and has the potential to benefit a wide range of applications beyond WSI analysis.

6. Additional Experiments and Analysis

6.1. Adaptability Analysis

To systematically evaluate our model's robustness and adaptability across different feature spaces, we conducted additional experiments using alternative feature extractors. While ResNet50 pretrained on ImageNet is the most widely adopted in computational pathology, recent self-supervised approaches have shown superior performance by pretraining on large-scale pathological datasets. In this section, we investigate how SPAN performs when integrated with other feature extractors. Specifically, we utilize PLIP [20], a vision-language foundation model designed for pathological image analysis, trained via contrastive learning on a large-scale pathological image-text dataset.

From Tables 1, 2, 5, and 6, we observe that most baseline methods achieve performance improvements when using PLIP as the feature extractor. For instance, ABMIL's accuracy on CAMELYON16 improves from 85.7% to 91.7%, and MHIM shows an improvement from 88.3% to 92.7%. There are only a few exceptions, such as PatchGCN which shows decreased performance on BRACS when using PLIP features (accuracy drops from 72.8% to 69.2%). Notably, our model demonstrates substantially larger performance gains across different datasets. On CAMELYON16, SPAN's accuracy improves by 3.4 percentage points, while on BRACS, it shows an improvement in accuracy from 72.5% to 75.7%). Not only for classification tasks, pathology-specific feature extractors also impact segmentation performance as shown in Tables 3 and 7. Although baseline methods generally improve with pathology-specific features, some exhibit inconsistent behavior. For example, GCN's Dice score on CAMELYON16 decreases from 0.841 to 0.755, demonstrating a lack of robustness across feature spaces. In contrast, SPAN consistently demonstrates substantial improvements across all datasets. Specifically, on CAMELYON17, SPAN's Dice score improves from 0.870 to 0.919. This consistent enhancement pattern across different feature representations further validates SPAN's architectural design principles in effectively preserving and leveraging spatial information for both classification and segmentation tasks.

These substantial improvements suggest that our model's feature interaction mechanism becomes more effective when the extracted features are more pathology-relevant, allowing SPAN to better leverage the rich semantic information captured by pathology-specific SSL pre-training.

Table 6. Classification performance on BRACS with PLIP as the feature extractor

Method	BRACS					
	Accuracy	AUC (Negative)	AUC (Atypical)	AUC (Positive)	Macro AUC	Macro F1
ABMIL backbone						
ABMIL	0.708 ± 0.022	0.886 ± 0.016	0.753 ± 0.060	0.924 ± 0.031	0.854 ± 0.090	0.584 ± 0.074
CLAM-SB	0.711 ± 0.023	0.876 ± 0.007	0.781 ± 0.044	0.921 ± 0.032	0.859 ± 0.072	0.623 ± 0.042
CLAM-MB	0.713 ± 0.041	0.880 ± 0.035	0.760 ± 0.048	0.912 ± 0.039	0.851 ± 0.080	0.622 ± 0.092
DTFD	0.720 ± 0.041	0.882 ± 0.034	0.740 ± 0.053	0.918 ± 0.045	0.847 ± 0.094	0.636 ± 0.077
DSMIL	0.725 ± 0.026	0.889 ± 0.013	0.781 ± 0.032	0.920 ± 0.034	0.863 ± 0.073	0.619 ± 0.074
MHIM	0.735 ± 0.026	0.885 ± 0.022	0.774 ± 0.047	0.924 ± 0.022	0.861 ± 0.078	0.636 ± 0.047
ACMIL	0.735 ± 0.049	0.888 ± 0.031	0.772 ± 0.042	0.932 ± 0.030	0.864 ± 0.083	0.636 ± 0.053
GNN backbone						
PatchGCN	0.692 ± 0.034	0.856 ± 0.039	0.697 ± 0.046	0.904 ± 0.041	0.819 ± 0.108	0.594 ± 0.060
TransMIL backbone						
TransMIL	0.730 ± 0.007	0.862 ± 0.010	0.690 ± 0.066	0.931 ± 0.013	0.828 ± 0.124	0.632 ± 0.060
RRT	0.733 ± 0.053	0.890 ± 0.015	0.756 ± 0.049	0.913 ± 0.038	0.853 ± 0.085	0.626 ± 0.082
SPAN backbone						
SPAN	0.757 ± 0.043	0.901 ± 0.034	0.817 ± 0.065	0.922 ± 0.033	0.880 ± 0.056	0.687 ± 0.058

Table 7. Segmentation performance on histopathology datasets

Method	CAMELYON16		CAMELYON17		SegCAMELYON		BACH	
	Dice	IoU	Dice	IoU	Dice	IoU	Dice	IoU
ABMIL	0.809 ± 0.021	0.679 ± 0.029	0.717 ± 0.087	0.565 ± 0.105	0.792 ± 0.052	0.659 ± 0.069	0.702 ± 0.147	0.557 ± 0.178
TransMIL	0.874 ± 0.011	0.776 ± 0.017	0.878 ± 0.054	0.786 ± 0.082	0.864 ± 0.035	0.762 ± 0.054	0.778 ± 0.112	0.648 ± 0.145
RRT	0.876 ± 0.012	0.779 ± 0.018	0.890 ± 0.032	0.803 ± 0.052	0.876 ± 0.054	0.783 ± 0.084	0.748 ± 0.122	0.609 ± 0.154
GCN	0.755 ± 0.070	0.611 ± 0.091	0.876 ± 0.024	0.779 ± 0.038	0.809 ± 0.068	0.684 ± 0.098	0.753 ± 0.121	0.615 ± 0.155
GAT	0.860 ± 0.015	0.754 ± 0.024	0.853 ± 0.038	0.746 ± 0.058	0.852 ± 0.066	0.747 ± 0.100	0.734 ± 0.158	0.598 ± 0.194
SPAN	0.900 ± 0.013	0.818 ± 0.021	0.919 ± 0.032	0.852 ± 0.053	0.884 ± 0.052	0.795 ± 0.084	0.814 ± 0.096	0.695 ± 0.132

† Method name indicates its corresponding architecture: ABMIL for MLP, TransMIL for vanilla Nystromformer, and RRT for region-based Nystromformer.

7. Visualization

To better understand how different models process WSIs with challenging characteristics, we visualize the raw attention weights across various architectures and transformer layers (Fig. 6). We specifically selected a challenging case with scattered tumor annotations to highlight the limitations of existing approaches and demonstrate how our proposed architecture addresses these challenges.

The visualizations reveal a critical limitation in conventional MIL approaches. Both ABMIL and RRTMIL exhibit highly skewed attention distributions, where the vast majority of weights are concentrated on an extremely limited number of tokens. This behavior undermines model robustness and is a direct consequence of softmax-based attention mechanisms operating over large token sets. As sequence length increases, even small differences in input logits are exponentially amplified when normalized, forcing attention to collapse onto a handful of tokens and causing decisions to depend heavily on a tiny subset of the input data.

SPAN alleviates this fundamental limitation through its hierarchical architecture. At the shallowest layer (Scale 0), SPAN begins subject to similar constraints as well. The key advantage emerges in deeper layers (Scales 1 and 2), where spatial downsampling progressively reduces token count. This reduction in the softmax denominator creates more balanced competition for attention weights allocation, preventing extreme concentration while still maintaining sufficient discriminative power. Instead of attention collapsing to just a few tokens, it remains distributed across multiple diagnostically relevant regions.

The combination of shifted window attention mechanisms with convolutional token condensation further enhances feature discrimination. As spatial resolution becomes coarser through the network, the attention mechanism effectively captures even small annotated tumor re-

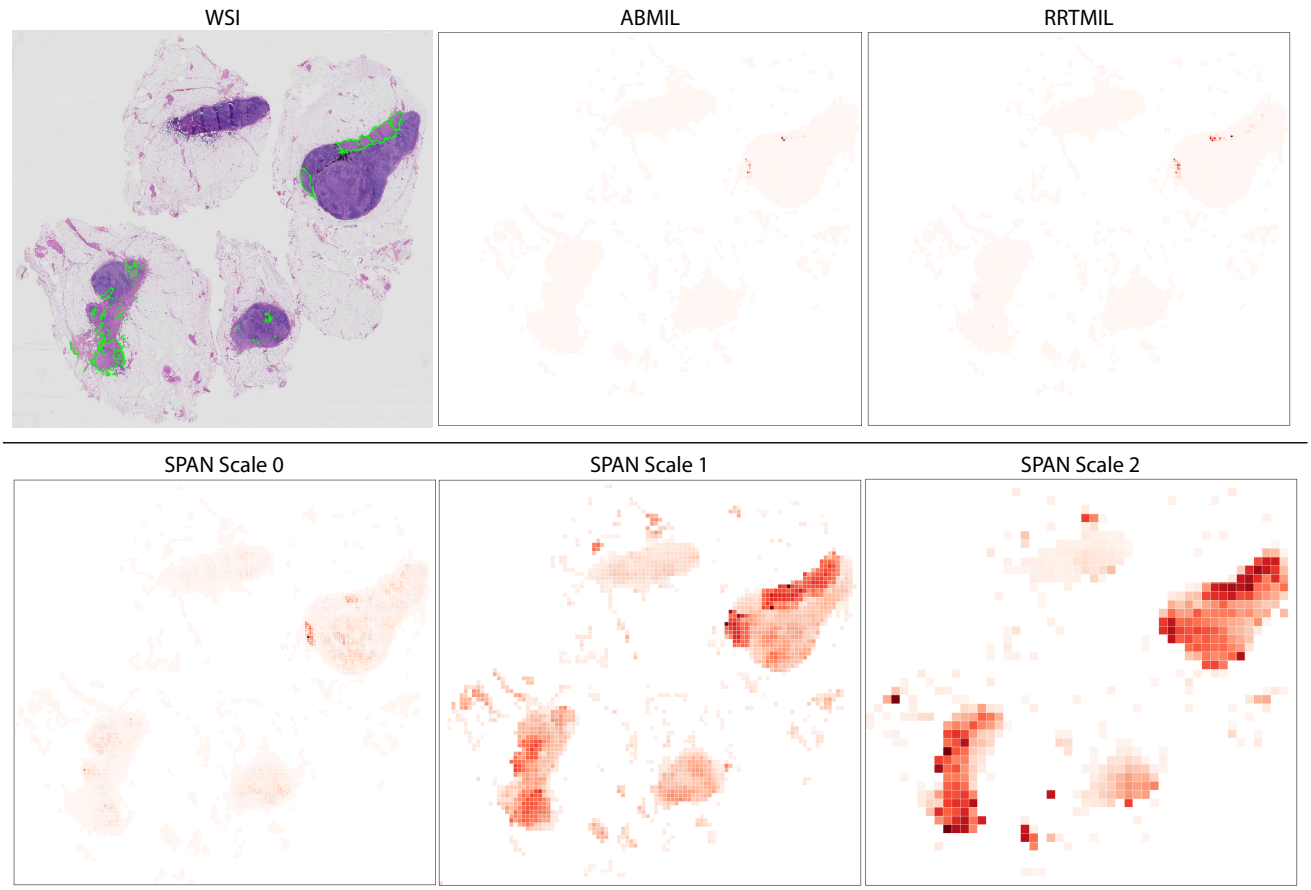

Figure 6. Visualization of raw attention weights across different models and scales. Top row: H&E stained WSI with tumor regions (green boxes), ABMIL attention maps, and RRTMIL attention maps. Bottom row: SPAN attention weights at different scales (0-2). All visualizations show unmodified post-softmax attention weights with red intensity indicating weight magnitude.

gions. Furthermore, tumor-relevant information propagate to adjacent tissue areas through both windowed attention mechanisms and convolutional operations, creating comprehensive contextual representations that consider both tumor regions and their surrounding environment.

This hierarchical design achieves an optimal balance between computational efficiency and preservation of diagnostic information. By systematically alleviating the limitations of softmax attention while maintaining multi-scale contextual awareness, SPAN delivers more robust decision-making without relying on the problematic few-token concentration seen in conventional approaches.