Learning Spatial-Preserving Hierarchical Representations for Digital Pathology

Supplementary Material

Contents of Supplementary Material

A. Implementation and Experimental Details

A.1. Task-specific Variants

A.1.1. SPAN-MIL: Slide-level Prediction

We utilize the global context tokens introduced in the CAR module for their comprehensive representations of the WSI across different scales. Let $\mathbf{h}_l^g \in \mathbb{R}^d$ denote the global context token from layer $l \in \{1, \ldots, L\}$. The slide-level representation is computed by:

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

The classification prediction is obtained through:

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

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.

A.1.2. SPAN-UNet: Patch-level Prediction

SPAN naturally extends to a U-Net [46] architecture through its hierarchical sparse design. The decoder maintains architectural symmetry with the encoder, using sparse deconvolution for upsampling in place of the downsampling operations.

Let $\{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_L\}$ denote the multi-scale feature maps from the encoder, where $\mathbf{H}_l \in \mathbb{R}^{N_l\times d}$ represents features at the l-th level.

The decoder generates features $\{\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_L\}$, processed at each stage through:

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

For the first decoding stage, $\mathbf{X}_1 = \mathbf{H}_L$. For subsequent stages, we implement skip connections by concatenating upsampled features with corresponding encoder features:

$$\mathbf{X}_l = \mathbf{G}_{l-1} \parallel \mathbf{H}_{L-l+1} \in \mathbb{R}^{N_l\times 2d}, \tag{12}$$

where $\parallel$ denotes feature concatenation. The final segmentation prediction at position i is:

$$\hat{y}_i = \text{softmax}(W^{\text{seg}}\mathbf{G}_L[i] + b^{\text{seg}}), \tag{13}$$

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.

A.2. Experimental Setup

A.2.1. Classification Datasets

WSI classification involves automatically categorizing tissues based on histopathological features, an essential process for accurate diagnosis, grading, and personalized treatment planning. We assessed SPAN's classification performance on three distinct diagnostic tasks, specifically tumor detection using the CAMELYON16 dataset [4], tumor grading employing the BRACS dataset [6], and HER2 biomarker status prediction using the Yale-HER2 dataset [22].

We followed the same strategy as above: all available slides were pooled, randomly shuffled, and split into training (∼70%), validation (∼15%), and test (∼15%). Experiments were repeated under five random seeds (0–4). All models were trained using cross-entropy loss. Model selection is based on validation set performance. Crucially, final predictions are made via direct class probability argmax, without any post-hoc threshold optimization, to better mirror real-world clinical deployment scenarios.

A.2.2. Segmentation Datasets

Slide-level segmentation requires precise pixel-level delineation of tumor regions, a challenging task crucial for diagnosis and prognosis. To rigorously evaluate SPAN's performance, we used fully annotated slides from multiple datasets: SegCAMELYON, Yale-HER2 [22], and

BACH [2]. To construct the SegCAMELYON benchmark, we curated tumor-positive slides from CAMELYON16 [4] and CAMELYON17 [3], applied exclusion masks to remove ambiguous regions, and consolidated the processed samples into a unified dataset.

All available slides were pooled, randomly shuffled, and split into training (∼70%), validation (∼10%), and test (∼20%). Experiments were repeated under five random seeds (0–4) to ensure robustness. Patches with over 20% tumor area are labeled positive for patch-level ground truth generation. For segmentation, we adopted 3-layer GCN and GAT models with 8-adjacent connectivity, following standard WSI analysis practices [11, 28, 57]. Model selection is based on validation set performance. Crucially, final predictions are made via direct class probability argmax, without any post-hoc threshold optimization, to better mirror real-world clinical deployment scenarios.

For segmentation training, we employed a hybrid loss that combines cross-entropy (CE) and Dice loss. Specifically, given the predicted probability map $\mathbf{p}$ and the ground-truth mask $\mathbf{y}$, we compute the standard pixel-wise CE loss $\mathcal{L}_{\text{CE}}(\mathbf{p}, \mathbf{y})$ and the Dice loss $\mathcal{L}_{\text{Dice}}(\mathbf{p}, \mathbf{y})$. The final objective is defined as:

$$\mathcal{L} = \begin{cases} (1-\lambda)\,\mathcal{L}_{\text{CE}} + \lambda\,\mathcal{L}_{\text{Dice}}, & \text{if } \sum \mathbf{y} > 0, \\ \mathcal{L}_{\text{CE}}, & \text{otherwise,} \end{cases}$$

where $\lambda = 0.75$ is the Dice weight. This design follows common practices in computer vision community, encouraging accurate boundary delineation when positives are present. All baseline methods were trained under this unified loss function for fair comparison.

A.2.3. Slide Preprocessing

Our preprocessing pipeline extends CLAM [43] by adding a grid alignment step, adjusting patch boundaries to the nearest multiple of 224 pixels for precise spatial coordinates.

To evaluate feature-space adaptability, we used two pretrained encoders to generate patch-level features from all datasets at 20x magnification. All patches were resized to 224×224 pixels prior to feature extraction. Our preprocessing pipeline addresses coordinate inconsistencies that arise from CLAM's background filtering mechanism. The original CLAM pipeline can generate patches with irregular starting coordinates due to tissue contour boundaries, making it difficult to establish consistent spatial relationships in a regular grid system. To resolve this, we introduced a grid alignment step that extends tissue contours to align with 224×224 pixel boundaries before patch extraction.

This alignment ensures that all patches map precisely to a regular grid coordinate system, eliminating potential rounding errors in spatial relationship modeling.

Algorithm 1: Expand Contours

```
global step_size = 224
def extend_contour(start_x, start_y, w, h):
    w += start_x % step_size
    h += start_y % step_size
    start_x -= start_x % step_size
    start_y -= start_y % step_size
    return start_x, start_y, w, h
# contour: (start_x, start_y, w, h)
contour = extend_contour(contour)
```

A.2.4. Patch Feature Extractor

In all experiments, the weights of these encoders were kept frozen to ensure a consistent feature extraction process.

ResNet50 As a standard baseline, we used a ResNet50 model pre-trained on ImageNet [26]. Following common practice in WSI analysis, we removed the final fully connected classification layer and used the output of the global average pooling layer. This process yields a 1024-dimensional feature vector for each patch, representing general-purpose visual features learned from natural images.

UNI We utilized UNI [13], a foundation model specifically tailored for computational pathology. Unlike general-purpose vision models, UNI is built upon a ViT-Large architecture and pretrained via self-supervised learning on a massive pan-cancer dataset comprising over 100 million tissue patches from more than 100,000 WSIs. This extensive domain-specific exposure enables the model to capture subtle histological patterns and high-level tissue semantics that are often missed by ImageNet supervised baselines.

B. Runtime

In the CAMELYON16 dataset, SPAN-MIL training runs 12.32 seconds per slide, compared to 3.09 seconds for AB-MIL and 16.96 seconds for TransMIL.

C. Experiments on Clinical and Biological Tasks

In addition to the human-annotated computer vision benchmarks presented in the main paper, we further evaluate SPAN on survival prediction, a clinical task that uses patient outcome supervision. Unlike traditional computer vision tasks where ground truth is defined by pathologists' visual perception, this task relies on objective biological signals from other modalities, including patient survival outcomes, which may not have obvious visual correlates. Consequently, it requires the model to discover complex, non-

Algorithm 2: SPAN Backbone with Rulebook Mechanism

Input: $\mathbf{P} \in \mathbb{N}^{N\times 2}$ (coordinates), $\mathbf{X} \in \mathbb{R}^{N\times d}$ (features)
Output: Refined features and global context
for *each layer in backbone* **do**
  // SAC Module: Sparse Convolution Rulebook
  $\mathbf{P}_{\text{out}} \leftarrow$ compute_output_coords($\mathbf{P}$, K, S, D)
  $\mathcal{R}_{\text{sparse}} \leftarrow$ build_sparse_rulebook($\mathbf{P}$, $\mathbf{P}_{\text{out}}$, $\mathcal{K}$)
  $\mathbf{X} \leftarrow$ execute_sparse_conv($\mathbf{X}$, $\mathcal{R}_{\text{sparse}}$, $\mathbf{W}$)
  // CAR Module: Sparse Attention Rulebook
  $\mathcal{W} \leftarrow$ generate_windows($\mathbf{P}_{\text{out}}$, window_size)
  $\mathcal{R}_{\text{local}} \leftarrow \{(i,j) \mid i,j \in w, \forall w \in \mathcal{W}\}$
  $\mathcal{R}_{\text{global}} \leftarrow \{(i, N+1), (N+1, i) \mid i \in [1, N]\}$
  $\mathbf{X} \leftarrow$ execute_attention($\mathbf{X}$, $\mathcal{R}_{\text{local}}$, $\mathcal{R}_{\text{global}}$)
  $\mathbf{P} \leftarrow \mathbf{P}_{\text{out}}$
return $\mathbf{X}$, *global_token*

Algorithm 3: Build Sparse Attention Rulebook

Input: $\mathbf{P} \in \mathbb{N}^{N\times 2}$ (coordinates), w (window size)
Output: $\mathcal{R}_{\text{local}}, \mathcal{R}_{\text{global}}$ (attention rulebooks)
// Create coordinate hash mapping
hash_ids $\leftarrow$ arange(1, $N+1$)
coord_transpose $\leftarrow$ $\mathbf{P}$.transpose()
spatial_bounds $\leftarrow$ (max(coord_transpose[0]) + 1, max(coord_transpose[1]) + 1)
coord_tensor $\leftarrow$ create_sparse_coo(coord_transpose, hash_ids, spatial_bounds)
index_matrix $\leftarrow$ coord_tensor.to_dense()
// Generate attention windows via spatial indexing
if *index_matrix.size()* $< 2w \times 2w$ **then**
  // Compact space: full attention
  spatial_indices $\leftarrow$ arange(num_elements)
  query_idx $\leftarrow$ spatial_indices.repeat_interleave(num_elements)
  key_idx $\leftarrow$ spatial_indices.repeat(num_elements)
else
  // Extended space: windowed attention
  window_blocks $\leftarrow$ generate_windows(index_matrix, w, mode)
  block_capacity $\leftarrow (2w)^2$
  intra_indices $\leftarrow$ arange(block_capacity)
  query_idx $\leftarrow$ intra_indices.unsqueeze(1).repeat(1, block_capacity).flatten()
  key_idx $\leftarrow$ intra_indices.repeat(block_capacity)
  query_hash $\leftarrow$ window_blocks.flatten()[query_idx]
  key_hash $\leftarrow$ window_blocks.flatten()[key_idx]
// Filter valid mappings and normalize hash indices
valid_mask $\leftarrow$ (query_hash $\neq 0$) $\wedge$ (key_hash $\neq 0$) $\wedge$ (query_hash $\neq$ key_hash)
$\mathcal{R}_{\text{local}} \leftarrow$ (query_hash[valid_mask] - 1, key_hash[valid_mask] - 1)
// Global context rulebook
$\mathcal{R}_{\text{global}} \leftarrow \{(\alpha, N+\beta), (N+\beta, \alpha) \mid \alpha \in [0, N-1], \beta \in [0, \text{num\_ctx} - 1]\}$
return $\mathcal{R}_{local}, \mathcal{R}_{global}$

trivial morphological patterns that are often subtle or invisible to the human eye. These experiments are complementary to the results in the main paper and demonstrate the applicability of SPAN beyond conventional vision benchmarks. We conducted 5 independent runs with different random seeds (0–4) using UNI features, where each run randomly splits the data into training/validation/test sets. For each run, we select the checkpoint that performs best on the validation set and report its corresponding test performance. We then report the mean and standard deviation across the 5 runs.

C.1. Survival Analysis

Survival prediction is a fundamental task in oncology that aims to estimate the risk of adverse events such as death or recurrence for each patient. Accurate risk stratification is crucial for personalized treatment planning. We evaluated SPAN for patient survival prediction on three TCGA cohorts: LGG (Lower-Grade Glioma), LUAD (Lung Adenocarcinoma), and LUSC (Lung Squamous Cell Carcinoma) [44].

For each cohort, we extracted clinical survival information including overall survival time and vital status. To prevent data leakage, we retained only one slide per patient by excluding duplicates based on Patient ID. We discretized the continuous survival times into $K = 3$ risk groups using quantile-based binning on uncensored patients, and then applied the resulting bins to all samples. Slides without valid survival annotations or with zero survival time were excluded from the analysis. For fair comparison with baseline methods, we adopted the same data split protocol: one-third of the data was reserved as the test set, and 15% of the remaining two-thirds was used for validation, with the rest allocated to training.

We trained the models using the negative log-likelihood survival loss [34], which accounts for both censored and

Algorithm 4: Spatial Window Indexing

Input: index_matrix, w (window radius), mode
Output: Active window blocks
$h, width \leftarrow$ index_matrix.size()
// Compute spatial alignment padding
row_align $\leftarrow (2w - h \bmod 2w) \bmod 2w$
col_align $\leftarrow (2w - width \bmod 2w) \bmod 2w$
if *row_align* > 0 *or col_align* > 0 **then**
  index_matrix $\leftarrow$ spatial_pad(index_matrix, alignment_spec, mode)
// Efficient spatial tessellation
window_tessellation $\leftarrow$ index_matrix.unfold(0, $2w$, $2w$).unfold(1, $2w$, $2w$)
// Filter active windows by occupancy
occupancy_map $\leftarrow$ window_tessellation.sum(dim=[-2, -1])
return *window_tessellation[occupancy_map* > 0*]*

Algorithm 5: Execute Rulebook-based Attention

Input: $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ (projections), $\mathcal{R}_{\text{local}}, \mathcal{R}_{\text{global}}$ (rulebooks)
Output: $\mathbf{H}_{\text{out}}$ (refined features)
// Local attention via spatial rulebook
for $(\alpha, \beta) \in \mathcal{R}_{local}$ **do**
  $\phi_{\alpha\beta} \leftarrow \frac{\mathbf{q}_\alpha^\top \mathbf{k}_\beta}{\sqrt{d}} + \mathcal{B}(\mathbf{P}[\alpha] - \mathbf{P}[\beta])$
$\mathbf{H}_{\text{local}} \leftarrow$ apply_rulebook_softmax($\{\phi_{\alpha\beta}\}, \mathbf{V}, \mathcal{R}_{\text{local}}$)
// Global attention via context rulebook
for $(\alpha, \beta) \in \mathcal{R}_{global}$ **do**
  $\psi_{\alpha\beta} \leftarrow \frac{\mathbf{q}_\alpha^\top \mathbf{k}_\beta}{\sqrt{d}}$
$\mathbf{H}_{\text{global}} \leftarrow$ apply_rulebook_softmax($\{\psi_{\alpha\beta}\}, \mathbf{V}, \mathcal{R}_{\text{global}}$)
$\mathbf{H}_{\text{out}} \leftarrow \mathbf{H}_{\text{local}} + \mathbf{H}_{\text{global}}$
return $\mathbf{H}_{out}$

uncensored events. The loss function is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[(1-c_i)(\log h_{y_i} + \log S_{y_i}) + c_i \log S_{y_i+1}\right], \tag{14}$$

where h represents the predicted discrete hazards, S is the survival probability, y_i is the discretized survival label, and c_i indicates censorship status (1 for censored, 0 for event observed).

Table 4 summarizes the results. SPAN consistently outperforms state-of-the-art MIL baselines across all three datasets. We attribute the sub-random performance of several baselines to our rigorous evaluation protocol, specifically the strict validation-based model selection and the discrete survival objective (using $K = 3$ risk groups). These constraints expose the tendency of standard MIL methods to overfit on limited data, whereas SPAN's hierarchical structure promotes robust performance.

Table 4. Survival prediction performance using UNI features. Baseline results are compared with SPAN-MIL. Values are mean C-index ± standard deviation.

Method	LGG	LUAD	LUSC
ACMIL	0.438 ± 0.092	0.460 ± 0.059	0.500 ± 0.048
ABMIL	0.416 ± 0.101	0.452 ± 0.070	0.500 ± 0.043
CLAM-MB	0.394 ± 0.088	0.455 ± 0.064	0.519 ± 0.046
CLAM-SB	0.436 ± 0.094	0.447 ± 0.074	0.528 ± 0.027
DSMIL	0.411 ± 0.104	0.471 ± 0.028	0.498 ± 0.081
RRTMIL	0.407 ± 0.094	0.476 ± 0.043	0.455 ± 0.049
TransMIL	0.419 ± 0.072	0.486 ± 0.030	0.488 ± 0.068
SPAN-MIL	0.647 ± 0.034	0.570 ± 0.044	0.584 ± 0.046

D. Potential Applications and Limitations

We believe SPAN provides a meaningful contribution to the digital pathology community. By adapting hierarchical vision architectures to the sparse and irregular structure of WSIs, SPAN establishes a robust and flexible computational foundation that aligns more closely with the intrinsic geometry of gigapixel pathology images. Beyond the benchmarks presented in this work, the framework naturally opens pathways for a broad range of advanced modeling strategies and clinical applications.

D.1. Foundation for Advanced Training Strategies

Although SPAN already achieves strong performance using a purely supervised objective, the architecture is well positioned to benefit from more sophisticated training schemes. Similar to how ABMIL has served as a general-purpose backbone for methods such as CLAM and ACMIL, SPAN can function as a versatile MIL foundation. Strategies such as knowledge distillation, curriculum-based hard negative mining, or task-specific auxiliary losses could be incorporated without altering the core design. Because SPAN preserves local spatial context and maintains stable hierarchical representations, these techniques may further enhance performance on challenging or fine-grained clinical tasks.

D.2. Large-scale Slide-level Pretraining

The hierarchical sparse design of SPAN is inherently well suited for large-scale whole-slide pretraining. Unlike patch-level pretraining paradigms that focus on isolated ROI features, SPAN preserves multi-resolution context and global tissue structure, making it compatible with emerging slide-level foundation model training [14, 19, 59]. By masking or

perturbing regions across the slide while retaining SPAN's spatial hierarchy, the model could learn robust and generalizable representations from large collections of unlabeled WSIs. Coupling SPAN with pathology reports or synthetic captions further enables slide–text alignment, similar to recent whole-slide vision and language models. Such pretraining strategies may substantially improve downstream performance, particularly for rare clinical conditions or limited-data settings where strong slide-level representations are essential.

D.3. Efficient Patch-level Extractor Adaptation

Although SPAN is currently trained with frozen patch-level features, the framework naturally supports parameter-efficient fine-tuning of the underlying foundation model. By inserting LoRA adapters [30] into a patch-level backbone such as UNI, one can selectively update the feature extractor while keeping the SPAN hierarchy fixed. This enables end-to-end optimization across both modules with minimal computational overhead. It provides a practical path for adapting large vision foundation models to domain-specific clinical tasks without the cost of full fine-tuning.

D.4. Complex Multimodal Tasks

Our results indicate that SPAN effectively captures both fine-grained spatial details and broader contextual relationships. This makes it a promising backbone for future vision and language modeling in computational pathology. Tasks such as report generation, captioning, or visual question answering require accurate grounding of visual features [9, 10, 14, 15], an area where current patch-based encoders often struggle due to limited positional structure. The spatially coherent representations produced by SPAN may therefore offer distinct advantages for multimodal reasoning over whole-slide images.

D.5. Limitations

Our work primarily establishes SPAN as a supervised learning baseline. We have not yet explored its integration with self-supervised slide-level pretraining, multimodal foundation models, or domain adaptation frameworks, all of which may further expand the model's utility. In addition, the current experiments only use single-scale inputs. Extending SPAN from single-scale to multi-scale inputs is a natural next step, and its hierarchical design may enable a more coherent integration of different magnifications than current isotropic multi-scale methods requiring multi-scale inputs, such as HIPT [12], H2MIL [28], and ZoomMIL [51]. Also, although the rulebook-based implementation is efficient, further optimization for specialized hardware accelerators such as GPUs with sparse kernels could improve speed for clinical deployment. Finally, our evaluation is limited to publicly available datasets. Assessing robustness across institutions, scanners, and staining protocols remains an important direction for future work.