Cancer Survival Analysis via Zero-shot Tumor Microenvironment Segmentation on Low-resolution Whole Slide Pathology Images

Jiao Tang, Wei Shao, Daoqiang Zhang*
The College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics
The Key Laboratory of Brain-Machine Intelligence Technology, Ministry of Education
tangjiao@nuaa.edu.cn, shaowei20022005@nuaa.edu.cn, dqzhang@nuaa.edu.cn

Abstract

The whole-slide pathology images (WSIs) are widely recognized as the golden standard for cancer survival analysis. However, due to the high-resolution of WSIs, the existing studies require dividing WSIs into patches and identify key components before building the survival prediction system, which is time-consuming and cannot reflect the overall spatial organization of WSIs. Inspired by the fact that the spatial interactions among different tumor microenvironment (TME) components in WSIs are associated with the cancer prognosis, some studies attempt to capture the complex interactions among different TME components to improve survival predictions. However, they require extra efforts for building the TME segmentation model, which involves substantial annotation workloads on different TME components and is independent to the construction of the survival prediction model. To address the above issues, we propose ZTSurv, a novel end-to-end cancer survival analysis framework via efficient zero-shot TME segmentation on low-resolution WSIs. Specifically, by leveraging tumor infiltrating lymphocyte (TIL) maps on the 50x down-sampled WSIs, ZTSurv enables zero-shot segmentation on other two important TME components (*i.e.,* tumor and stroma) that can reduce the annotation efforts from the pathologists. Then, based on the visual and semantic information extracted from different TME components, we construct a heterogeneous graph to capture their spatial intersections for clinical outcome prediction. We validate ZTSurv across four cancer cohorts derived from The Cancer Genome Atlas (TCGA), and the experimental results indicate that our method can not only achieve superior prediction results but also significantly reduce the computational costs in comparison with the state-of-the-art methods.

1 Introduction

Histopathology image analysis is a vital technology for cancer survival analysis [1, 2]. Traditional methods of survival analysis rely heavily on manual interpretation of these images by pathologists, which is time-consuming and prone to inter-observer variability. To address these challenges, computational approaches have been explored to assist the analysis process. Early computer-aided methods focus on handcrafted features extracted from specific regions of interests (ROIs), which are limited in scalability and generalization ability [3]. Recently, with the rapid development of the deep learning technology, training deep learning based whole-slide pathology image (WSI) analysis models for cancer survival prediction has gained significant attentions [4, 5]. However, the main challenge for survival analysis from the WSIs is that a high-resolution WSI is with large size (*e.g.,*

*Corresponding author

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

100,000-by-100,000 pixels), and thus it is impractical to directly feed them into deep neural networks due to memory limitations.

To make the analysis of WSIs memory-efficient, most of the existing studies firstly divide WSIs into multiple patches and identify key components before constructing the survival prediction model [6, 7, 8, 9, 10, 11]. Then, the patch-level representations are aggregated using attention [12] or pooling [13] strategies to predict the clinical outcome. However, dividing high-resolution WSIs into patches is computationally expensive, and the identified key components are insufficient to reflect the heterogeneous tumor microenvironment (TME) components and their spatial associations. As a highly heterogeneous disease, the progression of tumor is not only achieved by unlimited growth of the tumor cells, but also supported, stimulated, and nurtured by the TME components around (*i.e.,* stroma and lymphocyte) [14, 15].

For the purpose of capturing the spatial organizations among different TME components, the existing studies usually adopt graph neural networks (GNNs) to model the interactions among different TME components [16, 17, 18, 19, 20, 21, 22]. However, these methods require building the TME segmentation model for distinguishing different TME components before graph building, which need extra annotation efforts from the pathologists [3, 2, 23, 24]. Moreover, the existing studies train the TME segmentation and survival prediction models independently, which ignores the fact that the survival information could provide additional information to guide the TME segmentation task. For instance, we have more chance to see the TME component of lymphocyte near the tumor region for long survival patients, since it is widely recognized that the brisk interactions between lymphocyte and tumor regions will indicate a better clinical outcome [25]. Additionally, the existing GNN-based studies only extract visual features of each TME components as node representations while overlooking their corresponding semantic information, which limits the model's ability to distinguish different TME components for graph learning.

Based on the above considerations, in this paper, we propose ZTSurv, a novel end-to-end framework for cancer survival analysis through zero-shot segmentation of TME components on low-resolution WSIs. Specifically, instead of working on the high-resolution WSIs, ZTSurv leverages tumor infiltrating lymphocyte (TIL) maps on the 50x down-sampled WSIs to perform pixel-level zero-shot segmentation on other two important TME components (*i.e.,* tumor and stroma) through pathology-language foundation model (*i.e.,* PLIP). Then, based on the visual and semantic information extracted from different TME components, we construct a heterogeneous graph to capture their spatial intersections for clinical outcome prediction. Extensive experiments on four cancer cohorts derived from The Cancer Genome Atlas (TCGA) validate the effectiveness of ZTSurv, revealing its superior predictive performance and computational efficiency in comparison with the state-of-the-art approaches.

We summarize our main contributions as follows:

1. We propose a novel end-to-end framework ZTSurv for survival prediction of human cancers that can simultaneously segment different TME components and capture their spatial interactions for clinical outcome prediction.

2. We develop a zero-shot TME segmentation method that can leverage TIL maps to segment other two TME components (*i.e.,* tumor and stroma) by the aid of pathology-language foundation model, which can reduce the pixel-level annotation efforts on different types of TME components for constructing the semantic segmentation model.

3. Instead of working on the high-resolution WSIs, we implement our ZTSurv on the 50× down-sampled WSIs that can significantly reduce the computational cost in comparison with the existing studies.

4. We incorporate semantic information alongside the visual features to represent each TME component for graph construction, which can more effectively distinguish different TME components.

2 Related Work

2.1 Survival Analysis in Gigapixel WSIs

Gigapixel WSIs provide crucial insights for cancer prognosis but are challenging to process directly due to their large size [26]. Existing studies typically divide WSIs into multiple fixed-size patches and

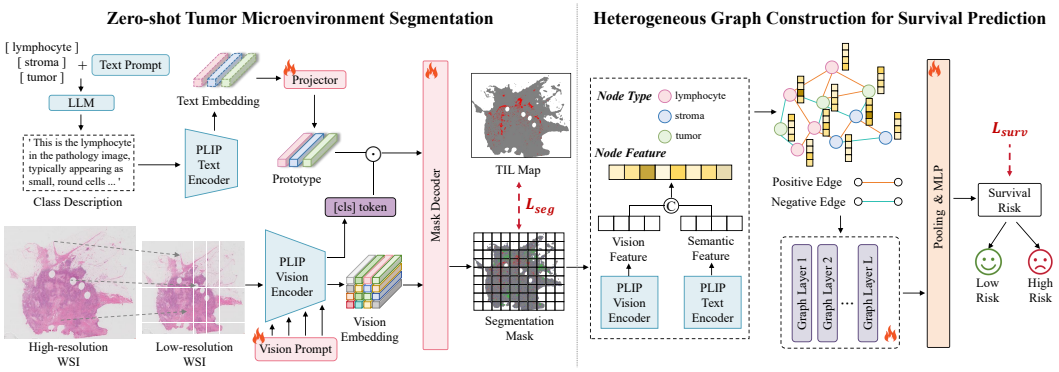


Figure 1: The overview of our proposed end-to-end framework ZTSurv for survival prediction of human cancers, which can simultaneously segment key TME components and capture their interactions for clinical outcome prediction.

extract useful information from these patches for survival prediction [6, 7, 27, 28, 8, 9, 11, 12, 13]. For instance, Mobadersany *et al.* [27] have presented a CNN-based survival prediction model based on the annotated ROIs extracted from WSIs. Zhu *et al.* [6] exploited and utilized all discriminative patches in WSIs to predict patients' survival status. Based on [6], Yao *et al.* [12] further employed an attention-based aggregation strategy to fuse informative patches for survival prediction. However, these methods fail to capture long-range spatial interactions of different patches in WSIs. To address the above challenges, a bunch of graph learning based survival prediction models are presented. For instance, Li *et al.* [21] proposed to model WSIs as graphs and then developed a graph convolutional neural network (graph CNN) with attention learning that better served the survival prediction by rendering the optimal graph representations of WSIs. Chen *et al.* [16] presented a spatially-resolved GCN [29] which hierarchically aggregated patch-level features to model local and global topological structures in WSIs. Di *et al.* [19] proposed a big-hypergraph factorization neural network to encode the correlation among vertices and hyperedges into two low-dimensional latent semantic spaces for better survival analysis. However, most of the graph learning studies overlook to discuss the interactions among different TME components that are important for cancer prognosis [25]. Also, they usually work on the high-resolution WSIs that will bring significant computational burden.

2.2 Capturing TME Heterogeneity for Survival Analysis

Recent studies have shown that capturing the heterogeneity of the TME is critical for improving the accuracy of survival prediction, as different TME components and their spatial interaction play important roles for patient outcome prediction [14, 30]. For instance, the studies in [31, 32, 23] have attempted to extract the TME information from WSIs for survival prediction. Han *et al.* [24] introduced a multi-scale heterogeneity-aware hypergraph representation framework to characterize the interactions between different TME components. Wu *et al.* [3] incorporated the concepts of prototype for TME analysis. Although these methods have shown promising results, they typically require dividing WSIs into patches, followed by training a dedicated segmentation or classification model, or fine-tuning a foundation model, to identify TME component types and select key patches before constructing graphs. However, such a patch pre-selection process relies heavily on additional annotations from pathologists to train a satisfactory classifier or segmentor, which is both labor-intensive and impractical when handling large volumes of high-resolution WSIs. What's more, they often neglect to incorporate semantic features during graph construction, limiting their ability to capture the biological significance of the interactions between different TME components.

3 Method

Fig. 1 presents the overview of our proposed ZTSurv. We leverage tumor infiltrating lymphocyte (TIL) maps on the 50x down-sampled WSIs to perform pixel-level zero-shot segmentation on other two important TME components (*i.e.,* tumor and stroma). The segmentation results are then used to construct a heterogeneous graph, where the different TME components are represented as nodes

based on their visual and semantic features. We capture the interactions between these components to form the graph edges. The constructed heterogeneous graph is updated through graph learning, which ultimately enables the prediction of survival outcomes.

3.1 Zero-shot TME Segmentation

Our objective is to segment TME tissue regions that are widely considered to be crucial for cancer prognosis (*e.g.,* lymphocyte [33], stroma [30], tumor [34]) on low-resolution WSIs using the corresponding TIL maps, where only the lymphocyte class is available as ground truth for training. To this end, we first down-sample the high-resolution WSI to match the size of the TIL map with width W' and height H'. We employ the PLIP [35] vision encoder, which divides the image into n patches with a patch size of p', and incorporate a learnable vision prompt to extract visual representations, including the [cls] token $\boldsymbol{g} \in \mathbb{R}^{d'}$ and vision embedding $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n\} \in \mathbb{R}^{n \times d'}$, where d' is the feature dimension of the PLIP model. Specifically, the input embeddings from the l-th multi-head attention (MHA) module of the ViT-based encoder in PLIP are represented as $\{\boldsymbol{g}^l, \boldsymbol{z}_1^l, \boldsymbol{z}_2^l, \ldots, \boldsymbol{z}_n^l\}$. We add a learnable vision prompt $\boldsymbol{P}^l = \{\boldsymbol{p}_1^l, \boldsymbol{p}_2^l, \ldots, \boldsymbol{p}_m^l\}$ into the input and the l-th MHA module processes the tokens as follows (detailed illustrations are available in the *Appendix* C):

$$[\boldsymbol{g}^l, \boldsymbol{Z}^l, \_] = Layer^l[\boldsymbol{g}^{l-1}, \boldsymbol{Z}^{l-1}, \boldsymbol{P}^{l-1}]. \tag{1}$$

Instead of using generic text prompts like "a photo of {}" , we generate class-specific textual descriptions $\hat{\boldsymbol{H}}$ that capture the appearance attributes of various TME components using a large language model (LLM) (details in *Appendix* D). These descriptions are encoded via the PLIP text encoder to obtain text embeddings $\boldsymbol{H} \in \mathbb{R}^{C \times d'}$, where C is the number of classes. A projector ψ_p, composed of three linear layers, maps the text embeddings to prototypes $\boldsymbol{E} = \psi_p(\boldsymbol{H}) \in \mathbb{R}^{C \times d'}$. Inspired by [36], we adopt a mask decoder consisting of three layers of lightweight transformers to segment the TME components at the pixel level. The input $\boldsymbol{Q}(query), \boldsymbol{K}(key), \boldsymbol{V}(value)$ are computed as:

$$\boldsymbol{Q} = \psi_q(\boldsymbol{E} \odot \boldsymbol{g}) \in \mathbb{R}^{C \times d'}, \quad \boldsymbol{K} = \psi_k(\boldsymbol{Z}) \in \mathbb{R}^{n \times d'}, \quad \boldsymbol{V} = \psi_v(\boldsymbol{Z}) \in \mathbb{R}^{n \times d'}, \tag{2}$$

where $\odot$ is the Hadamard product, ψ_q, ψ_k, and ψ_v represent linear transformations. At each transformer layer, the query $\boldsymbol{Q}$ is updated to better capture the semantic correlations with the visual embeddings. To obtain the predicted mask, we calculate the score map using scaled dot-product attention from the final layer with a Sigmoid activation to ensure that the segmentation results of each class are independently generated:

$$\text{ScoreMap} = \text{Sigmoid}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}) \in \mathbb{R}^{C \times n}, \tag{3}$$

where $\sqrt{d_k}$ is the dimension of the keys as a scaling factor. The score map is then reshaped to $C \times (H'/p') \times (W'/p')$, and the final segmentation mask $\boldsymbol{M} \in \mathbb{R}^{H' \times W'}$ is obtained by applying an Argmax operation along the class dimension followed by an up-sampling step to restore the original spatial resolution. We train the zero-shot segmentation model using focal loss [37] and dice loss [38, 39]. Notably, only the lymphocyte class, which is visible in the TIL map, contributes to the loss calculation, while other classes are ignored.

3.2 Heterogeneous Graph Construction

To explicitly model the heterogeneity and spatial organization of the TME, we construct a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$ based on the predicted segmentation mask $\boldsymbol{M}$, where $\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}$ represent the set of entities (vertices or nodes), the set of relations (edges), the set of entity types, the space of edge attributes, respectively. Each node $v \in \mathcal{V}$ is associated with a type through a mapping function $\tau(v) \in \mathcal{A}$. An edge $e = (s, r, t) \in \mathcal{E}$ connects a source node s to a target node t, where the edge type is given by $\phi(e) = r \in \mathcal{R}$. Each node v has a d-dimensional feature vector $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X}$ represents the embedding space for node feature. Specifically, we divide the down-sampled WSI into a set of non-overlapping patches using a fixed window size $s' \times s'$. For each patch, we determine its tissue label $y \in \mathcal{A}$ based on the dominant class within the patch area. Patches labeled

as background are discarded, and the remaining patches are retained as graph nodes $\mathcal{V}$. To construct node features $\mathcal{X}$, we first extract visual features $\boldsymbol{f}^{\text{vis}} \in \mathbb{R}^{d_v}$ for each patch using the PLIP vision encoder. In addition, considering the semantic gap and inherent heterogeneity among different TME components, we further obtain semantic embeddings $\boldsymbol{f}^{\text{text}} \in \mathbb{R}^{d_t}$ by feeding the corresponding class name into the PLIP text encoder. Then, the final node representation can be calculated as:

$$\boldsymbol{x} = [\boldsymbol{f}^{\text{vis}} \parallel \boldsymbol{f}^{\text{text}}] \in \mathbb{R}^{d_v+d_t}, \quad \boldsymbol{x} \in \mathcal{X}. \tag{4}$$

Here, $\parallel$ denotes the concatenation operation. The node types and node features reflect the biological roles of different TME components and preserve both visual and semantic heterogeneity at the node level. Based on the defined nodes and their feature representations, we establish edges and assign edge attributes to capture neighborhood relationships. For each node $v \in \mathcal{V}$, we apply the k-nearest neighbor algorithm to identify the k most similar nodes to it, and create directed edges connecting v to each of its neighbors. For each edge $e \in \mathcal{E}$, we compute the Pearson correlation coefficient $u \in \mathcal{U}$ between the feature vectors of the source and target nodes as its continuous attribute, where $\mathcal{U}$ represents the set of continuous edge attributes. The edge type $r \in \mathcal{R}$ is labeled as "positive" if the coefficient is positive and "negative" otherwise. The edge attributes introduce heterogeneity at the relational level and help to highlight implicit meta-relations among different tissue regions in the WSI. To reduce potential noise introduced by uncertain or spurious correlations, we apply data augmentation strategies during training, including random edge and feature dropout.

3.3 Heterogeneous Graph Learning for Survival Analysis

Traditional graph attention mechanisms fail to effectively address the heterogeneity inherent in the graph structure [3]. To address this challenge, we incorporate node features that capture both visual representations and type-specific information, alongside continuous edge attributes, into the aggregation process, which allows the model to capture the complex interactions and diverse relationships between different TME components.

Edge Updating. For each edge $e \in \mathcal{E}$, we project its continuous attributes $u^{l-1} \in \mathcal{U}$ from the $(l-1)$-th graph learning layer to the l-th layer $u^l = W_{edge}u^{l-1}$ using a linear projector W_{edge}.

Node Updating. Instead of using edge similarity as weights for node updates, we account for the heterogeneity inherent in both edges and nodes (detailed illustrations are available in the *Appendix* F). Specifically, in each layer, we update the embedding of a node $v \in \mathcal{V}$ by aggregating information from all its neighbors. We refer to node v as target node t, with its neighbors denoted as $\mathcal{V}(t) = \{s_1, s_2, \ldots, s_N\}$, where $\mathcal{V}(t)$ represents the set of source nodes that point to the target node t, and N is the number of neighbors. The set of edges associated with node t is denoted as $\mathcal{E}(t) = \{e_1, e_2, \ldots, e_N\}$. At layer l, for each (s_i, e_i, t) and attention head h, we first project the target node t into a query vector $\boldsymbol{F}^h_{\text{query}}$ using a linear projector $\boldsymbol{W}^h_{\tau(t)}$, and the source node into key and value vectors $\boldsymbol{F}^h_{\text{key},i}$ and $\boldsymbol{F}^h_{\text{value},i}$ using $\boldsymbol{W}^h_{\tau(s_i)}$:

$$\boldsymbol{F}^h_{\text{query}} = \boldsymbol{W}^h_{\tau(t)}\boldsymbol{x}^{l-1}_t, \quad \boldsymbol{F}^h_{\text{key},i} = \boldsymbol{W}^h_{\tau(s_i)}\boldsymbol{x}^{l-1}_{s_i}, \quad \boldsymbol{F}^h_{\text{value},i} = \boldsymbol{W}^h_{\tau(s_i)}\boldsymbol{x}^{l-1}_{s_i}, \tag{5}$$

where $\boldsymbol{x}^{l-1}_{s_i}$ and $\boldsymbol{x}^{l-1}_t$ represent the node features of the source node s_i and target node t at the $(l-1)$-th layer, and $\tau(\cdot) \in \mathcal{A}$ is a mapping function that assigns the corresponding type to each node. Then we calculate the attention score for each edge e_i on h-th attention head using the query vector $\boldsymbol{F}^h_{\text{query}}$ and the key vector $\boldsymbol{F}^h_{\text{key},i}$ modulated by the edge's continuous attribute u_i, and apply the Softmax function across all edges in $\mathcal{E}(t)$ to obtain the normalized attention scores:

$$att^h(t, e_i) = \boldsymbol{F}^h_{\text{key},i} \cdot u_i \cdot \boldsymbol{F}^h_{\text{query}} / \sqrt{d_v + d_t}, \tag{6}$$

$$w^h(t, e) = \underset{\forall e \in \mathcal{E}(t)}{\text{Softmax}}(att^h(t, e)), \tag{7}$$

where $\sqrt{d_v + d_t}$ is the scaling factor to ensure numerical stability, $w^h(t, e) \in \mathbb{R}^N$ represents the final attention score of the edges associated with target node t on h-th attention head. Finally, the updated

embedding of the target node t is computed by aggregating the weighted value vectors:

$$\boldsymbol{x}_t = \sum_{i=1}^{N} (\|_{\ h\in[1,H]} w^h(t, e_i) \cdot \boldsymbol{F}^h_{\text{value},i}), \tag{8}$$

where $\|_{\ h\in[1,H]}$ denotes the concatenation of all attention heads. The aggregation process results in an updated feature that effectively combines both node-type and edge-attribute information, thereby capturing the graph's heterogeneity. After completing the L-th layer of graph learning, we employ a global attention based pooling [40] to dynamically calculate a weighted sum of the node features in the graph, transforming them into a WSI-level embedding representation. The representation is then passed through a MLP to predict the final survival risk score. For the k'-th patient, we can model the survival function $f^{(k')}_{surv}(T \geq t, D^{(k')})$ and hazard function $f^{(k')}_{hazard}(T = t|T \geq t, D^{(k')})$ given the relative clinical information $D^{(k')} = (X^{(k')}, c^{(k')}, t^{(k')})$, where $X^{(k')}$ represents the patient's WSI, $c^{(k')} \in \{0, 1\}$ indicates the censoring status, and $t^{(k')} \in \mathbb{R}^+$ denotes the overall survival time. After graph learning, we learn the representation $f(D^{(k')})$ and compute the survival loss $\mathcal{L}_{surv}(\{f(D^{(k')}), t^{(k')}, c^{(k')}\}^{N_D}_{k'=1})$ of all patients, where N_D is the number of samples in the training set. Specifically, we adopt the negative log-likelihood (NLL) loss [41] to quantify the difference between the predicted survival risk and the actual clinical outcomes (details can be found in the *Appendix* B):

$$\mathcal{L}_{surv} = -\sum_{k'=1}^{N_D} c^{(k')} log(f^{(k')}_{surv}(t|f(D^{(k')}))) \tag{9}$$

$$+ (1 - c^{(k')}) log(f^{(k')}_{surv}(t-1|f(D^{(k')}))) \tag{10}$$

$$+ (1 - c^{(k')}) log(f^{(k')}_{hazard}(t|f(D^{(k')}))). \tag{11}$$

3.4 Overall Loss

The overall loss consists of the zero-shot TME segmentation loss $\mathcal{L}_{seg}$ and the survival loss $\mathcal{L}_{surv}$, which can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{seg} + \gamma \mathcal{L}_{surv} = \alpha \mathcal{L}_{focal} + \beta \mathcal{L}_{dice} + \gamma \mathcal{L}_{surv}, \tag{12}$$

where $\{\alpha, \beta, \gamma\}$ are coefficients that balance the contributions of different losses. In *Appendix* H, we further analysis the time complexity of ZTSurv.

4 Experiments

4.1 Datasets

We conduct experiments on four cancer cohorts: Breast Invasive Carcinoma (BRCA) (1,016 cases), Uterine Corpus Endometrial Carcinoma (UCEC) (520 cases), Lung Adenocarcinoma (LUAD) (540 cases), and Bladder Urothelial Carcinoma (BLCA) (369 cases). All WSIs for these cancer types are sourced from The Cancer Genome Atlas (TCGA) repository [42] [2]. The corresponding 50x down-sampled tumor-infiltrating lymphocyte (TIL) maps are obtained from [43] [3].

4.2 Experimental Settings

Implementation Details. For each cancer cohort, we evaluate model performance using a 5-fold cross-validation strategy. We use openslide [44] tool to process and down-sample the high-resolution WSIs. The WSI segmentation is implemented with the open-source MMSegmentation toolbox [45] with PyTorch 1.10.1. We employ the pre-trained PLIP ViT-B/32 model to extract visual and textual features, with a feature dimension of 512. GPT-4 [46] is used as the LLM to generate class

[2] https://portal.gdc.cancer.gov/

[3] https://www.cancerimagingarchive.net/analysis-result/til-wsi-tcga/

Table 1: Comparisons of C-index (mean ± std) for survival prediction with SOTA methods over four cancer datasets. The best and the second-best results are highlighted in **bold** and underlined. p. represents methods sampling patches from WSIs; g. denotes graph-based methods; t. refers to methods considering TME heterogeneity; w. represents methods without splitting WSIs into patches.

Model	Design	BRCA	UCEC	LUAD	BLCA	Overall
CLAM-SB [51]	p.	0.581 ± 0.041	0.550 ± 0.113	0.549 ± 0.053	0.563 ± 0.049	0.561
CLAM-MB [51]	p.	0.541 ± 0.119	0.551 ± 0.118	0.556 ± 0.067	0.601 ± 0.070	0.562
Co-Pilot [28]	p.	0.544 ± 0.076	0.557 ± 0.112	0.571 ± 0.083	0.587 ± 0.056	0.564
DeepAttnMISL [12]	p.	0.598 ± 0.066	0.639 ± 0.068	0.601 ± 0.041	0.597 ± 0.028	0.609
DSMIL [11]	p.	0.548 ± 0.080	0.581 ± 0.164	0.538 ± 0.047	0.552 ± 0.052	0.555
WSISA [6]	p.	0.514 ± 0.071	0.539 ± 0.114	0.575 ± 0.055	0.581 ± 0.050	0.552
DeepGraphSurv [21]	p.+g.	0.587 ± 0.033	0.622 ± 0.097	0.595 ± 0.011	0.584 ± 0.037	0.597
HGSurvNet [22]	p.+g.	0.624 ± 0.093	0.614 ± 0.034	0.587 ± 0.045	0.578 ± 0.041	0.601
PatchGCN [16]	p.+g.	0.608 ± 0.043	0.678 ± 0.128	0.614 ± 0.007	0.599 ± 0.034	0.625
H2GT [24]	p.+g.+t.	0.618 ± 0.032	0.671 ± 0.096	0.629 ± 0.046	0.623 ± 0.058	0.635
TMEGL [2]	p.+g.+t.	0.623 ± 0.017	0.700 ± 0.053	0.631 ± 0.023	0.627 ± 0.029	0.645
ProtoSurv [3]	p.+g.+t.	0.625 ± 0.009	0.705 ± 0.131	**0.638 ± 0.026**	0.629 ± 0.043	0.649
ZTSurv (ours)	w.+g.+t.	**0.642 ± 0.029**	**0.726 ± 0.113**	0.637 ± 0.033	**0.637 ± 0.042**	**0.661**

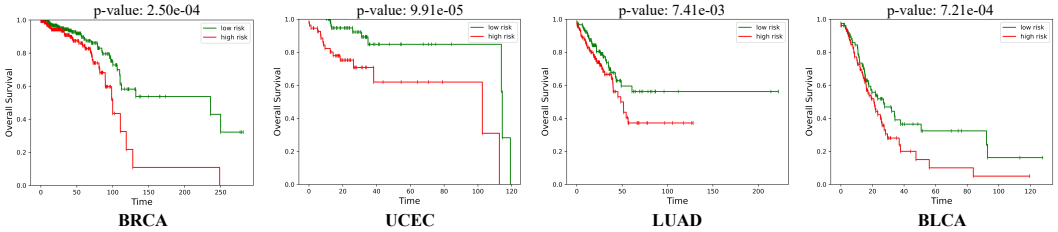


Figure 2: Kaplan–Meier curves for predicted high-risk (red) and low-risk (green) groups across four cancer datasets. A p-value < 0.05 indicates statistical significance.

descriptions for different tissue categories. For graph construction, we set the window size $s' = 64$ and select $k = 8$ neighbors per node. Data augmentation is applied by randomly dropping node and edge features with a dropout rate of 0.2. The layer for graph learning L is set to 2, with $H = 4$ attention heads. The hyperparameters for the loss function are set as $\alpha = 20$, $\beta = 1$, and $\gamma = 20$. We use the AdamW optimizer [47] with a learning rate of 2×10^{-5} and a weight decay of 1×10^{-5}. The batch size is set to 8, and we train the model for 2K iterations. A detailed parameter analysis is provided in the *Appendix* H.5.

Evaluation Metrics. To evaluate TME segmentation performance, we use the mean of class-wise Intersection over Union (mIoU) on lymphocyte, *i.e.,* the only TME component with available ground truth annotations, to measure the overlap between predicted and reference regions. To assess predictive performance, we use the concordance index (C-index) [48] for evaluating the ability to correctly rank the survival risk of different patients. For qualitative assessment, we employ Kaplan-Meier curves [49] with log-rank test [50] to visualize patient stratification, distinguishing between low and high-risk patients with two separate survival distributions. More details can be found in *Appendix* G.

4.3 Comparison with State-Of-The-Art Methods.

We compare our proposed method with several state-of-the-art approaches for survival prediction: (1) CLAM-SB [51], (2) CLAM-MB [51], (3) Co-Pilot [28], (4) DeepAttnMISL [12], (5) DSMIL [11], (6) WSISA [6], (7) DeepGraphSurv [21], (8) HGSurvNet [22], (9) PatchGCN [16], (10) H2GT [24], (11) TMEGL [2], (12) ProtoSurv [3]. Among them, CLAM-SB, CLAM-MB, Co-Pilot, DeepAttnMISL, DSMIL, and WSISA are classical WSI survival prediction approaches; DeepGraphSurv, HGSurvNet, and PatchGCN model spatial relationships by constructing homogeneous graphs for survival prediction; H2GT, TMEGL, and ProtoSurv leverage the heterogeneity of the TME for survival analysis. All

Table 2: Comparisons of mIoU (mean ± std) for TME segmentation with SOTA methods over four cancer datasets. The best and the second-best results are highlighted in **bold** and underlined.

Model	BRCA	UCEC	LUAD	BLCA	Overall
ZegFormer [52]	0.536 ± 0.109	0.459 ± 0.125	0.566 ± 0.021	0.561 ± 0.047	0.531
ZegCLIP [53]	0.547 ± 0.105	0.475 ± 0.121	0.572 ± 0.040	0.570 ± 0.055	0.541
TagCLIP [54]	0.541 ± 0.117	0.474 ± 0.141	0.568 ± 0.036	0.579 ± 0.059	0.541
ZTSurv-Seg	0.552 ± 0.096	0.478 ± 0.150	0.579 ± 0.050	0.589 ± 0.056	0.550
ZTSurv (ours)	**0.574 ± 0.023**	**0.506 ± 0.147**	**0.595 ± 0.023**	**0.608 ± 0.055**	**0.571**

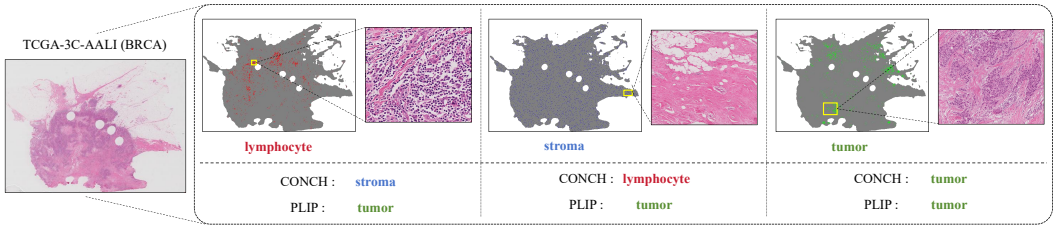


Figure 3: Zero-shot TME segmentation comparison of ZTSurv, CONCH, and PLIP.

these methods require splitting WSIs into patches as an essential processing step, while ZTSurv can avoid this step and directly utilize low-resolution WSIs for training an end-to-end survival prediction model.

As shown in Table 1, ZTSurv outperforms all classical [51, 28, 12, 11, 6] and graph-based [21, 22, 16] methods by a margin of 1.8%-4.8%, demonstrating the effectiveness of our method that incorporates TME heterogeneity for more accurate survival prediction. Compared to recent TME-aware models such as H2GT [24], TMEGL [2], and ProtoSurv [3], ZTSurv still achieves the highest C-index on 3 out of 4 cancer datasets, with an improvement of 0.8%-2.1%. These results highlight the strength of ZTSurv that can simultaneously realize the TME segmentation and survival prediction tasks, while the existing studies treat these two tasks independently that miss the inherent correlation among them.

4.4 Patient Stratification

We perform Kaplan–Meier analysis by separating patients into high-risk and low-risk groups based on predicted risk scores, using the median value within each validation set as the cut-off. The log-rank test [50] is then applied to compute p-values that assess the statistical significance of survival differences between the two groups, with smaller p-values indicating better stratification. In Fig. 2 and *Appendix* H.1, we present the stratification ability of our method across four cancer datasets and compare it with two best competitors (*i.e.,* ProtoSurv and TMEGL). The results clearly show that our method achieves more distinct separation between high-risk (red) and low-risk (green) groups across all cohorts with lower p-values, which further demonstrates the effectiveness of our method.

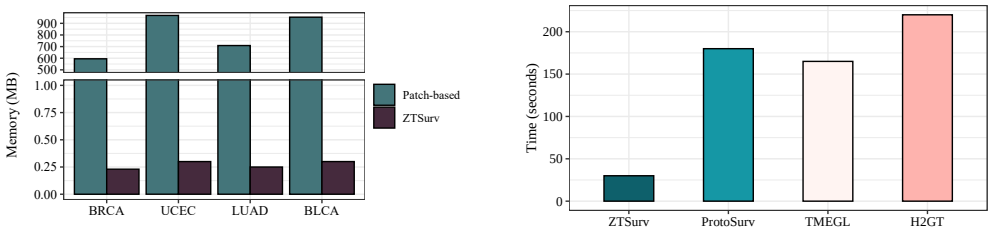
4.5 Ablation Study and Analysis

Component Ablation. To further investigate the contribution of each component in ZTSurv, we conduct an ablation study by removing or replacing key modules in *Appendix* H.2: 1) w/o text prompt: Removing text prompt of zero-shot segmentation. 2) w/o vision prompt: Removing vision prompt of zero-shot segmentation. 3) w/o node type: Constructing a homogeneous graph without considering node types during graph construction. 4) w/o semantic feature: Without considering semantic feature for node representation during graph construction. 5) w/o vision feature: Without considering vision feature for node representation during graph construction. 6) w/o edge attribute: Without considering edge attribute during graph construction and learning. As can be observed from Table 4 in *Appendix*, ZTSurv is superior to its variants, indicating that each component of our method is effective in improving survival prediction performance.

Comparison of Zero-shot TME Segmentation. We compare the zero-shot TME segmentation performance with several state-of-the-art methods: (1) ZegFormer [52], (2) ZegCLIP [53], (3)

Table 3: Effect of tissue category choices on C-index (mean ± std) over four cancer datasets.

	BRCA	UCEC	LUAD	BLCA	Overall
lymphocyte + tumor	0.628 ± 0.020	0.713 ± 0.128	0.629 ± 0.06	0.629 ± 0.047	0.650
lymphocyte + stroma	0.634 ± 0.043	0.704 ± 0.056	0.633 ± 0.065	0.623 ± 0.049	0.649
lymphocyte + tumor + stroma	0.642 ± 0.029	0.726 ± 0.113	0.637 ± 0.033	0.637 ± 0.042	0.661



(a) Average Memory Usage per WSI.

(b) Average Inference Time per WSI.

Figure 4: Comparison of Average Memory Usage and Inference Time per WSI.

TagCLIP [54]. We also evaluate a variant of our method, ZTSurv-seg, which is trained solely with segmentation loss and without survival-guided information. As shown in Table 2, ZTSurv consistently outperforms all compared methods across four cancer datasets, achieving the highest overall mIoU of 0.571. Additionally, compared to its variant ZTSurv-seg, which lacks survival guidance, ZTSurv shows consistent gains in segmentation performance across all cohorts. These results suggest that incorporating survival supervision can provide informative signals that help refine the segmentation process, resulting in more prognostically meaningful TME delineation. In Fig. 3, we further present the visualization results of segmentation and compare it with CONCH [55] and PLIP [35]. As shown in Fig. 3, our method accurately captures key TME components, while existing survival analysis approaches that rely on pathology foundation models, like CONCH and PLIP, often struggle to identify non-tumor components, limiting their ability to fully capture TME heterogeneity. The results further highlight the effectiveness of our approach in zero-shot TME segmentation, enabling more comprehensive and precise survival analysis. In the *Appendix* H.4, we provide more visualization results on other three datasets (*i.e.,* BLCA, LUAD, and UCEC).

Comparison of Survival Prediction with Different Zero-shot Classifiers. In *Appendix* H.3, we compare ZTSurv with three alternative approaches for the zero-shot TME segmentation stage (*i.e.,* CONCH [55], PLIP [35], and a UNI classifier [56] finetuned as described in [3]) for survival outcome prediction. As shown in Fig. 9 in *Appendix*, ZTSurv consistently achieves superior C-index scores across all four cancer datasets, demonstrating that our method achieves better performance.

Effect of Different Tissue Categories. In our work, we consider three key TME components, (*i.e.,* lymphocyte, tumor, and stroma), to capture the complex TME. To further investigate the impact of different tissue categories on survival prediction, we conduct experiments with different combinations of tissue components (*i.e.,* lymphocyte + tumor, lymphocyte + stroma) to evaluate their influence on survival analysis. As shown in Table 3, including a more comprehensive set of tissue categories (*i.e.,* lymphocyte + tumor + stroma) consistently achieves the highest C-index across four datasets, indicating that capturing a broader TME leads to better survival prediction.

Time and Memory Analysis. Compared with conventional patch-based methods that divide high-resolution WSIs into thousands of tiles, ZTSurv offers significant advantages in both time and memory efficiency. As shown in Fig. 4, on a single NVIDIA RTX 4090 GPU, ZTSurv reduces average memory usage per WSI by approximately 3,000 times and decreases average inference time by about 6 times compared to other methods. This is primarily because ZTSurv directly processes the down-sampled WSI without the need to generate and store thousands of small patches, significantly reducing computational overhead and memory requirements, which further demonstrates its superior scalability and efficiency for large-scale whole-slide image analysis.

5 Conclusion

In this paper, we propose ZTSurv, a novel end-to-end framework for cancer survival analysis that integrates survival prediction with zero-shot TME segmentation on low-resolution WSIs. ZTSurv eliminates the need for manual annotations by leveraging TIL maps to segment key TME components, and introduces survival-guided segmentation to enhance the identification of prognostically TME regions. By performing segmentation directly on 50× down-sampled WSIs, the framework significantly reduces computational cost. Furthermore, the incorporation of semantic features in graph construction allows the model to better capture complex tissue interactions. Extensive experiments on four TCGA cohorts demonstrate the effectiveness and efficiency of ZTSurv, indicating its strong potential for scalable and practical histopathology analysis.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62136004, 62272226, 62102188), Key Research and Development Plan of Jiangsu Province, China under Grant BE2022842.

References

[1] Xinliang Zhu, Jiawen Yao, and Junzhou Huang. Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 544–547. IEEE, 2016.

[2] Wei Shao, YangYang Shi, Daoqiang Zhang, JunJie Zhou, and Peng Wan. Tumor microenvironment interactions guided graph learning for survival analysis of human cancers from whole-slide pathological images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11694–11703, 2024.

[3] Junxian Wu, Xinyi Ke, Xiaoming Jiang, Huanwen Wu, Youyong Kong, and Lizhi Shao. Leveraging tumor heterogeneity: Heterogeneous graph representation learning for cancer survival prediction in whole slide images. *Advances in Neural Information Processing Systems*, 37:64312–64337, 2024.

[4] Yawen Wu, Michael Cheng, Shuo Huang, Zongxiang Pei, Yingli Zuo, Jianxin Liu, Kai Yang, Qi Zhu, Jie Zhang, Honghai Hong, et al. Recent advances of deep learning for computational histopathology: principles and applications. *Cancers*, 14(5):1199, 2022.

[5] Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep neural network models for computational histopathology: A survey. *Medical image analysis*, 67:101813, 2021.

[6] Xinliang Zhu, Jiawen Yao, Feiyun Zhu, and Junzhou Huang. Wsisa: Making survival prediction from whole slide histopathological images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7234–7242, 2017.

[7] Richard J Chen, Ming Y Lu, Wei-Hung Weng, Tiffany Y Chen, Drew FK Williamson, Trevor Manz, Maha Shady, and Faisal Mahmood. Multimodal co-attention transformer for survival prediction in gigapixel whole slide images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4025, 2021.

[8] Weiyi Wu, Chongyang Gao, Joseph DiPalma, Soroush Vosoughi, and Saeed Hassanpour. Improving representation learning for histopathologic images with cluster constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21404–21414, 2023.

[9] Ming Y Lu, Tiffany Y Chen, Drew FK Williamson, Melissa Zhao, Maha Shady, Jana Lipkova, and Faisal Mahmood. Ai-based pathology predicts origins for cancers of unknown primary. *Nature*, 594(7861):106–110, 2021.

[10] Richard J Chen, Chengkuan Chen, Yicong Li, Tiffany Y Chen, Andrew D Trister, Rahul G Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16144–16155, 2022.

[11] Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2021.

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

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

[14] Amanda J Oliver, Peter KH Lau, Ashleigh S Unsworth, Sherene Loi, Phillip K Darcy, Michael H Kershaw, and Clare Y Slaney. Tissue-dependent tumor microenvironments and their impact on immunotherapy responses. *Frontiers in immunology*, 9:70, 2018.

[15] Andreas Heindl, Sidra Nawaz, and Yinyin Yuan. Mapping spatial heterogeneity in the tumor microenvironment: a new era for digital pathology. *Laboratory investigation*, 95(4):377–384, 2015.

[16] 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–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pages 339–349. Springer, 2021.

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

[18] Jiangbo Shi, Lufei Tang, Yang Li, Xianli Zhang, Zeyu Gao, Yefeng Zheng, Chunbao Wang, Tieliang Gong, and Chen Li. A structure-aware hierarchical graph-based multiple instance learning framework for pt staging in histopathological image. *IEEE Transactions on Medical Imaging*, 42(10):3000–3011, 2023.

[19] Donglin Di, Jun Zhang, Fuqiang Lei, Qi Tian, and Yue Gao. Big-hypergraph factorization neural network for survival prediction from whole slide image. *IEEE Transactions on Image Processing*, 31:1149–1160, 2022.

[20] Yu Zhao, Fan Yang, Yuqi Fang, Hailing Liu, Niyun Zhou, Jun Zhang, Jiarui Sun, Sen Yang, Bjoern Menze, Xinjuan Fan, et al. Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4837–4846, 2020.

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

[22] Donglin Di, Changqing Zou, Yifan Feng, Haiyan Zhou, Rongrong Ji, Qionghai Dai, and Yue Gao. Generating hypergraph-based high-order representations of whole-slide histopathological images for survival prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5800–5815, 2022.

[23] Tsai Hor Chan, Fernando Julio Cendra, Lan Ma, Guosheng Yin, and Lequan Yu. Histopathology whole slide image analysis with heterogeneous graph representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15661–15670, 2023.

[24] Minghao Han, Xukun Zhang, Dingkang Yang, Tao Liu, Haopeng Kuang, Jinghui Feng, and Lihua Zhang. Multi-scale heterogeneity-aware hypergraph representation for histopathology whole slide images. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024.

[25] Yingli Zuo, Yawen Wu, Zixiao Lu, Qi Zhu, Kun Huang, Daoqiang Zhang, and Wei Shao. Identify consistent imaging genomic biomarkers for characterizing the survival-associated interactions between tumor-infiltrating lymphocytes and tumors. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 222–231. Springer, 2022.

[26] Navid Farahani, Anil V Parwani, and Liron Pantanowitz. Whole slide imaging in pathology: advantages, limitations, and emerging perspectives. *Pathology and Laboratory Medicine International*, pages 23–33, 2015.

[27] Pooya Mobadersany, Safoora Yousefi, Mohamed Amgad, David A Gutman, Jill S Barnholtz-Sloan, José E Velázquez Vega, Daniel J Brat, and Lee AD Cooper. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proceedings of the National Academy of Sciences*, 115(13):E2970–E2979, 2018.

[28] Ramin Nakhli, Allen Zhang, Ali Mirabadi, Katherine Rich, Maryam Asadi, Blake Gilks, Hossein Farahani, and Ali Bashashati. Co-pilot: Dynamic top-down point cloud with conditional neighborhood aggregation for multi-gigapixel histopathology image representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21063–21073, 2023.

[29] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[30] Roy M Bremnes, Tom Dønnem, Samer Al-Saad, Khalid Al-Shibli, Sigve Andersen, Rafael Sirera, Carlos Camps, Inigo Marinez, and Lill-Tove Busund. The role of tumor stroma in cancer progression and prognosis: emphasis on carcinoma-associated fibroblasts and non-small cell lung cancer. *Journal of thoracic oncology*, 6(1):209–217, 2011.

[31] Sajid Javed, Arif Mahmood, Muhammad Moazam Fraz, Navid Alemi Koohbanani, Ksenija Benes, Yee-Wah Tsang, Katherine Hewitt, David Epstein, David Snead, and Nasir Rajpoot. Cellular community detection for tissue phenotyping in colorectal cancer histology images. *Medical image analysis*, 63:101696, 2020.

[32] Henrik Failmezger, Sathya Muralidhar, Antonio Rullan, Carlos E de Andrea, Erik Sahai, and Yinyin Yuan. Topological tumor graphs: a graph-based spatial model to infer stromal recruitment for immunosuppression in melanoma histology. *Cancer research*, 80(5):1199–1209, 2020.

[33] J Galon, MC Dieu-Nosjean, E Tartour, C Sautes-Fridman, WH Fridman, et al. Immune infiltration in human tumors: a prognostic factor that should not be ignored. *Oncogene*, 29(8):1093–1102, 2010.

[34] Clifton F Mountain. New prognostic factors in lung cancer: biologic prophets of cancer cell aggression. *Chest*, 108(1):246–254, 1995.

[35] 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.

[36] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, et al. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35:4971–4982, 2022.

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[38] N Fausto Milletari and Ahmadi Seyed-Ahmad V-Net. Fully convolutional neural networks for volumetric medical image segmentation.

[39] Rongjian Zhao, Buyue Qian, Xianli Zhang, Yang Li, Rong Wei, Yang Liu, and Yinggang Pan. Rethinking dice loss for medical image segmentation. In *2020 IEEE international conference on data mining (ICDM)*, pages 851–860. IEEE, 2020.

[40] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

[41] Shekoufeh Gorgi Zadeh and Matthias Schmid. Bias in cross-entropy-based training of deep survival networks. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3126–3137, 2020.

[42] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

[43] Joel Saltz, Rajarsi Gupta, Le Hou, Tahsin Kurc, Pankaj Singh, Vu Nguyen, Dimitris Samaras, Kenneth R Shroyer, Tianhao Zhao, Rebecca Batiste, et al. Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. *Cell reports*, 23(1):181–193, 2018.

[44] Adam Goode, Benjamin Gilbert, Jan Harkes, Drazen Jukic, and Mahadev Satyanarayanan. Openslide: A vendor-neutral software foundation for digital pathology. *Journal of pathology informatics*, 4(1):27, 2013.

[45] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020.

[46] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[47] Zhenxun Zhuang, Mingrui Liu, Ashok Cutkosky, and Francesco Orabona. Understanding adamw through proximal methods and scale-freeness. *arXiv preprint arXiv:2202.00089*, 2022.

[48] Frank E Harrell Jr, Kerry L Lee, and Daniel B Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387, 1996.

[49] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[50] Nathan Mantel et al. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemother Rep*, 50(3):163–170, 1966.

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

[52] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11583–11592, 2022.

[53] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11175–11185, 2023.

[54] Jingyao Li, Pengguang Chen, Shengju Qian, Shu Liu, and Jiaya Jia. Tagclip: improving discrimination ability of zero-shot semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[55] Ming Y Lu, Bowen Chen, Drew FK Williamson, Richard J Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, et al. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024.

[56] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3):850–862, 2024.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our abstract and introduction accurately reflect our paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss our limitations at Appendix H.7.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: We do not have theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide detailed descriptions of our experimental setup in section 4.2.

   Guidelines:

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

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

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

Answer: [No]

Justification: The dataset is accessible online. We will release our code in the official version of the subsequent paper to provide reproduction and provide more help to the future community.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide detailed descriptions of the hyperparameter and optimizer setup in section 4.2.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We provide the standard deviation in our experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide information on the computer resources in section 4.5.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: All the data we used comes from public datasets, and there are no violations of NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss potential societal impacts in appendix I.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: We do not release data and pretrained models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The assets used in our paper are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: All the data we used comes from public datasets.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: All the data we used comes from public datasets.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We used GPT-4 as the LLM to generate detailed text descriptions for different tissue categories.

Guidelines:

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

A Overview

In this appendix, we first provide detailed analysis of survival prediction in B. We then provide the detailed illustrations of vision prompt and text prompt for zero-shot TME segmentation in C and D, respectively. In E and F, we present the detailed illustrations of mask decoder of zero-shot TME segmentation and node updating in graph learning, respectively. Then, we provide the evaluation metric for survival prediction in G. In H, we present more experiments and analysis, including comparisons of patient stratification H.1, ablation study of main components H.2, comparisons of survival prediction with different zero-shot classifiers H.3, more zero-shot TME segmentation visualization results H.4, parameter analysis H.5, time complexity analysis H.6, and discussions of limitations and future work H.7. Finally, we discuss the potential ethical issues which may arise in our study in I.

B Survival Analysis

Survival prediction aims to model the time until an event occurs, typically formulated as an ordinal regression problem. In clinical data, the event (*e.g.,* death) is not always observed due to censoring, such as when a patient is lost to follow-up. These right-censored cases introduce uncertainty, as we only know that the event did not occur before the last recorded time.

Following our notation in Sec. 3.3, let $D = (X, c, t)$ represent the patient's clinical data, where X is patient's pathology image, $t \in \mathbb{R}^+$ denotes the overall survival time, $c \in \{0, 1\}$ specifies whether the data is right-censored. We denote T as a continuous random variable representing survival time. The survival function $f_{surv}(T \geq t, D)$ estimates the probability of a patient surviving beyond time t, while the hazard function $f_{hazard}(t|D) = f_{hazard}(T = t|T \geq t, D)$ quantifies the risk of the event occurring at time t given survival up to that point, which is defined as:

$$f_{hazard}(T = t) = \lim_{\partial t \to 0} \frac{P(t \leq T \leq t + \partial t \mid T \geq t)}{\partial t}, \tag{13}$$

which can be used to estimate $f^{(k')}_{surv}(T \geq t, D^{(k')})$ by integrating the hazard function f_{hazard}. The most common model for learning hazard functions is the Cox Proportional Hazards (CoxPH) model, where the hazard is modeled as:

$$f_{hazard}(t|D) = f^0_{hazard}(t) \exp(\theta^\top D). \tag{14}$$

Here, $f^0_{hazard}(t)$ is the baseline hazard, and θ contains parameters that modulate risk based on the input features D. In deep learning applications, θ is typically produced by the final layer of a neural network and is optimized using the partial log-likelihood of the Cox model via stochastic gradient descent.

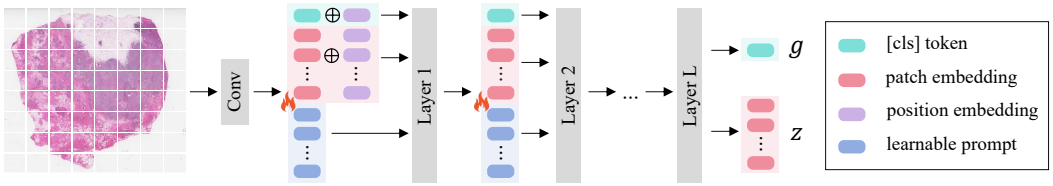


Figure 5: Illustration of vision prompt for zero-shot TME segmentation.

C Vision Prompt of Zero-shot TME Segmentation

In Sec. 3.1, we use the vision prompt, as shown in Fig. 5, to enhance the extraction of visual features from low-resolution WSIs. The [cls] token, patch embeddings, and learnable prompts are passed through each layer, with only the learnable prompts being trainable.

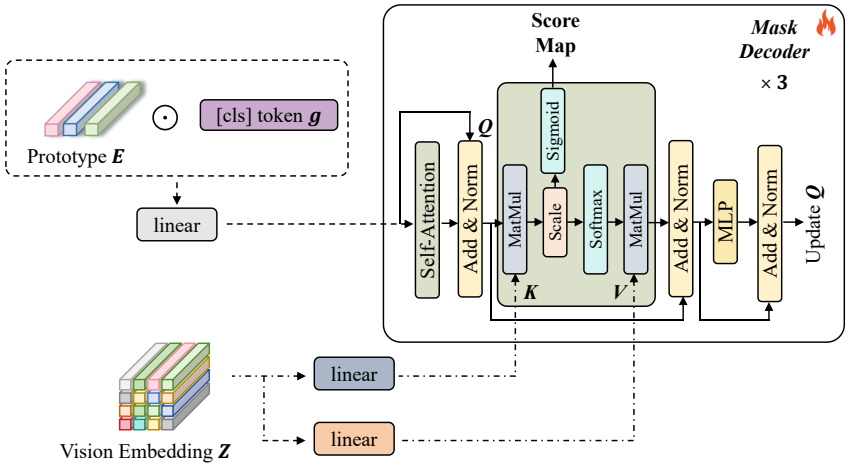


Figure 6: A detailed illustration of mask decoder in zero-shot TME segmentation.

D Text Prompt of Zero-shot TME Segmentation

In Sec. 3.1, we generate class-specific textual descriptions which capture the appearance attributes of TME components using the LLM. Specifically, instead of using generic prompts like *"a photo of {}"*, we use the following prompts fed into the LLM: *"Describe {} in the pathology image in detail, including features such as color and shape"*. We list the textual descriptions for key TME components (*i.e.,* lymphocyte, stroma, tumor) as follows:

- "lymphocyte": This is the lymphocyte in the pathology image, typically appearing as small, round cells with a light blue cytoplasm and dark purple nuclei.
- "stroma": This is the stroma in the pathology image, typically appearing as fibrous tissue in pale pink or beige tones, supporting the surrounding cells.
- "tumor": This is the tumor in the pathology image, typically appearing as irregularly shaped cells with darker purple or blue cytoplasm and enlarged, pleomorphic nuclei.

E Detailed Illustrations of Mask Decoder

In Fig. 6, we present a detailed illustration of the mask decoder inspired by [36] in zero-shot TME segmentation, which progressively refines feature representations to generate accurate segmentation masks.

F Detailed Illustrations of Node Updating

In Fig. 7, we provide a detailed illustration of node updating in graph learning, where the target node is solely associated with source nodes A and B in this example.

G Evaluation Metric for Survival Analysis

We use the concordance index (C-index) to evaluate the predictive accuracy of survival analysis models, which evaluates a model's ability to correctly rank pairs of samples based on their predicted risk of experiencing an event (*e.g.,* death) within a given time frame. Specifically, samples are sorted by their predicted survival scores, and the C-index reflects the proportion of correctly ordered pairs. The metric is defined as:

$$C\text{-}index = \frac{1}{N'(N'-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbb{I}(T_i < T_j)(1 - c_j), \tag{15}$$

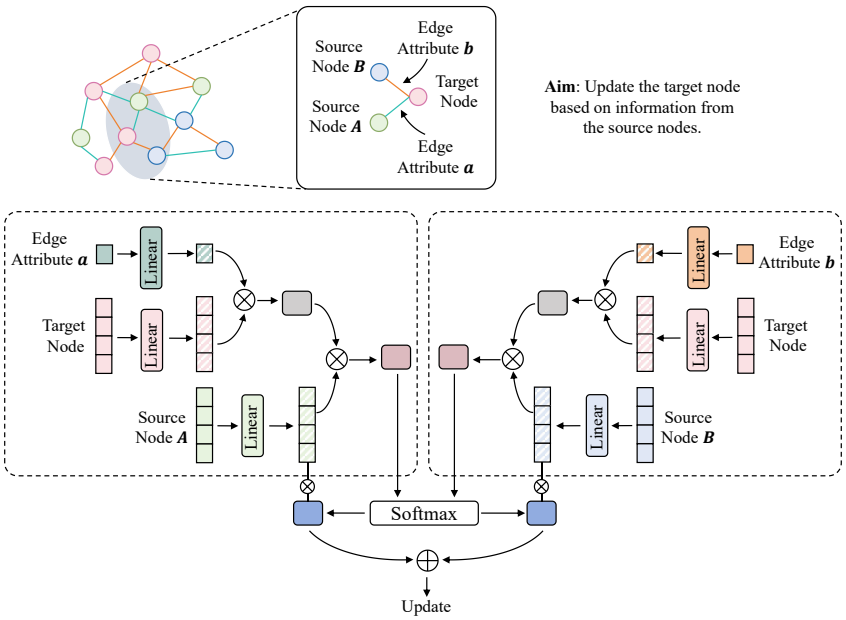


Figure 7: A detailed illustration of node updating in graph learning.

where N' is the total number of patients, and $\mathbb{I}$ is the indicator function that returns 1 when the condition holds and 0 otherwise.

H More Experiments and Analysis

H.1 Patient Stratification

In addition to evaluating prognostic performance using the C-index, patient stratification is another key aspect of cancer survival analysis, enabling the identification of subgroups with distinct clinical outcomes for personalized treatment. We compare the patient stratification ability of ZTSurv with its two best competitors (*i.e.,* ProtoSurv and TMEGL) in Fig. 8, and the results demonstrate that ZTSurv consistently achieves clearer separation between risk groups, indicating more accurate prognosis.

H.2 Component Ablation

Table 4 shows the ablation results of ZTSurv on C-index across four cancer datasets. As shown in Table 4, removing either the text prompt or vision prompt in TME segmentation results in a noticeable drop in performance across all datasets, with the overall C-index decreasing to 0.644 and 0.650 respectively, confirming the effectiveness of prompt-based guidance for zero-shot segmentation. In terms of graph construction, we further examine the influence of node types and feature representations. It is clear that removing node type information, *i.e.,* treating all nodes as homogeneous, leads to a noticeable performance drop, underscoring the importance of preserving semantic distinctions among TME regions. Furthermore, excluding the vision feature results in the most significant degradation, indicating that visual cues derived from WSIs are critical for accurate survival prediction. The absence of semantic features also degrades performance, suggesting their complementary value in capturing contextual information. Additionally, eliminating edge attributes weakens the model's ability to capture spatial relationships, resulting in a further drop in the overall C-index to 0.651. The results emphasize the critical role of each component in our framework.

H.3 Comparison of Survival Prediction with Different Zero-shot Classifiers

In Fig. 9, we compare ZTSurv with three alternative approaches for the zero-shot TME segmentation stage (*i.e.,* CONCH [55], PLIP [35], and a UNI classifier [56] finetuned as described in [3]) to predict

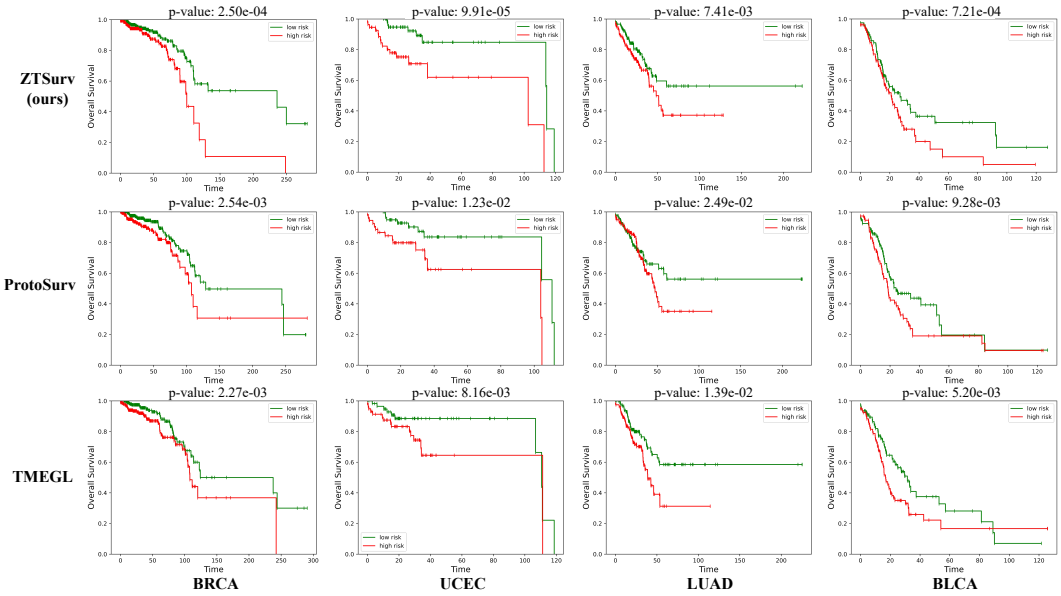


Figure 8: Kaplan–Meier curves for predicted high-risk (red) and low-risk (green) groups across four cancer datasets under different comparative methods. A p-value < 0.05 indicates statistical significance.

Table 4: Ablation study of ZTSurv on C-index (mean ± std) over four cancer datasets.

	BRCA	UCEC	LUAD	BLCA	Overall
w/o text prompt	0.618 ± 0.021	0.706 ± 0.098	0.631 ± 0.035	0.622 ± 0.018	0.644
w/o vision prompt	0.621 ± 0.016	0.723 ± 0.089	0.623 ± 0.026	0.631 ± 0.027	0.650
w/o node type	0.615 ± 0.025	0.702 ± 0.110	0.624 ± 0.019	0.630 ± 0.043	0.643
w/o semantic feature	0.624 ± 0.023	0.701 ± 0.071	0.635 ± 0.015	0.624 ± 0.026	0.646
w/o vision feature	0.611 ± 0.026	0.669 ± 0.075	0.625 ± 0.040	0.621 ± 0.023	0.632
w/o edge attribute	0.632 ± 0.024	0.707 ± 0.072	0.632 ± 0.036	0.632 ± 0.006	0.651
ZTSurv	0.642 ± 0.029	0.726 ± 0.113	0.637 ± 0.033	0.637 ± 0.042	0.661

the survival outcome. The results further demonstrate the effectiveness of our method in capturing TME heterogeneity.

H.4 More Zero-shot TME Segmentation Visualization

In Fig. 10, we provide more zero-shot TME segmentation visualization results on BLCA, LUAD, and UCEC cancer datasets. The results demonstrate that ZTSurv can capture key TME components more accurately.

H.5 Parameter Analysis

We analyze the sensitivity of the number of neighbors k used for connecting nodes and the window size s' for node generation. As shown in Table 5, $k = 8$ yields the best overall performance, while both smaller ($k = 4$) and larger ($k = 16$) values result in performance degradation. For the window size s', we observe that $s' = 64$ achieves the highest C-index across all datasets, while smaller window sizes result in overly complex graphs with redundant information and larger sizes fail to capture sufficient local detail, which highlights the importance of balancing graph density and information granularity in node construction.

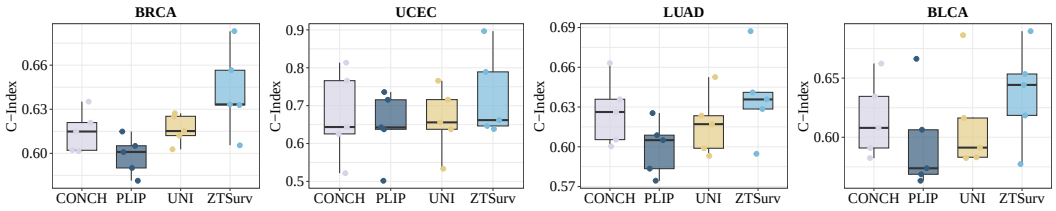


Figure 9: Comparisons of C-index (mean ± std) with different TME segmentation methods over four cancer datasets.

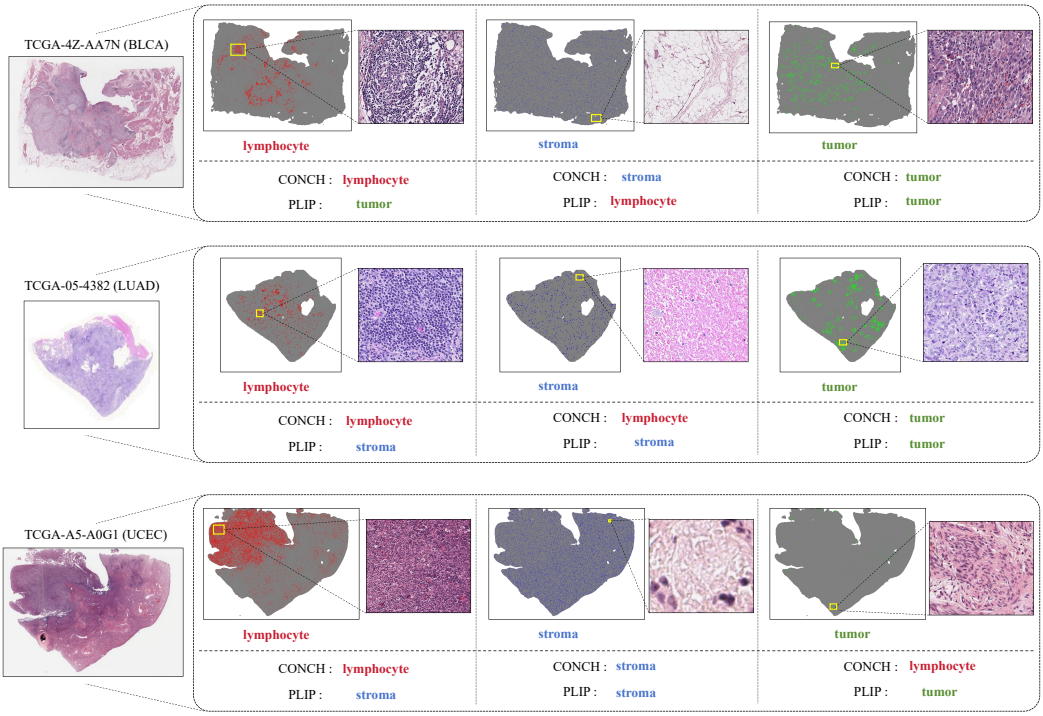


Figure 10: Zero-shot TME segmentation comparison of ZTSurv, CONCH, and PLIP.

H.6 Time Complexity Analysis

The complexity of ZTSurv is dominated by three components. Firstly, zero-shot TME segmentation on the down-sampled WSI has a complexity of $O(H'W'd')$. Secondly, the complexity of constructing a heterogeneous graph with n' nodes is $O(n'(d_v + d_t))$. Finally, the graph neural network performs message passing over L layers, with a per-layer cost of $O(n'k(d_v + d_t))$. In summary, the time complexity of our ZTSurv is $O(H'W'd') + O(n'(d_v + d_t)) + O(Ln'k(d_v + d_t)) = O(H'W'd' + n'(1 + Lk)(d_v + d_t))$.

H.7 Limitations and Future Work

In this work, we focus on capturing key TME components, including lymphocyte, tumor, and stroma, for survival prediction. However, as demonstrated in Table 3, including a broader range of TME components can potentially lead to better predictive performance. In future work, we will explore more diverse tissue types, such as blood vessels, necrosis, and fibroblasts, to capture a more comprehensive representation of the TME and further enhance model robustness.

Table 5: Parameter analysis of C-index (mean ± std) over four cancer datasets.

	BRCA	UCEC	LUAD	BLCA	Overall
$k = 4$	0.633 ± 0.037	0.703 ± 0.051	0.636 ± 0.047	0.632 ± 0.045	0.651
$k = 8$	0.642 ± 0.029	0.726 ± 0.113	0.637 ± 0.033	0.637 ± 0.042	0.661
$k = 16$	0.630 ± 0.022	0.698 ± 0.063	0.622 ± 0.049	0.625 ± 0.041	0.644
$s' = 32$	0.635 ± 0.022	0.702 ± 0.073	0.632 ± 0.052	0.634 ± 0.057	0.651
$s' = 64$	0.642 ± 0.029	0.726 ± 0.113	0.637 ± 0.033	0.637 ± 0.042	0.661
$s' = 128$	0.635 ± 0.037	0.695 ± 0.038	0.631 ± 0.037	0.611 ± 0.024	0.643

I Ethical Discussions

Ethical Considerations. The Cancer Genome Atlas (TCGA) dataset used in this study is a publicly available resource widely utilized in pathology research. Given its open nature and established use, its application in this study does not present significant ethical concerns. The TIL maps employed were generated based on publicly available data without the involvement of any identifiable patient information, ensuring no individuals are adversely impacted. As such, this study adheres to ethical guidelines without compromising privacy or rights.

Potential Positive Social Impacts. The proposed method has the potential to improve patient outcomes by enabling more accurate and comprehensive analysis of tumor microenvironments, facilitating early diagnosis and personalized treatment planning. Moreover, it can reduce the workload of pathologists by automating routine analyses, potentially increasing the scalability and efficiency of cancer diagnosis in clinical practice.

Potential Negative Social Impacts. As this work focuses on cancer survival prediction, it is important to acknowledge potential social impacts, including but not limited to:

- Diagnostic Errors. Like all AI-based methods, this approach is not immune to errors. Incorrect predictions or misclassifications could have serious consequences for patient care and treatment decisions. Therefore, these tools should serve as decision aids, complementing but not replacing human medical judgment.
- Privacy Concerns. WSI datasets can contain sensitive information, and the leakage of such data may pose significant privacy risks to patients. To mitigate this, our study exclusively relies on publicly available datasets where personal identifiers are either absent or appropriately protected.