Robust Multimodal Survival Prediction with Conditional Latent Differentiation Variational AutoEncoder

Supplementary Material

6. Detailed Methods

6.1. Survival Analysis

For *k-th* patient, we can model the survival and hazard functions given $I^{(k)} = (P^{(k)}, G^{(k)}, c^{(k)}, t^{(k)})$, where $P^{(k)}$ represents the set of whole slide images, $G^{(k)}$ denotes the genomic profiles, $c^{(k)} \in \{0, 1\}$ indicates the right uncensorship status, and $t^{(k)} \in \mathbb{R}^+$ is the overall survival time (in months). The hazard function $f^{(k)}_{hazard}(T = t|T \geq t, I^{(k)})$ measures the instantaneous risk of death at time point k for the *k-th* patient, which can be defined as:

$$f^{(k)}_{hazard}(T = t) = \lim_{\partial t \to 0} \frac{P(t \leq T \leq t + \partial t | T \geq t)}{\partial t} \quad (16)$$

The survival function $f^{(k)}_{surv}(T \geq t, I^{(k)})$ quantifies the probability of surviving after a specified time t, which can be estimated via the cumulative hazard function $f^{(k)}_{hazard}(T = t|T \geq t, I^{(k)})$ as follows:

$$f_{surv}(T \geq t, I^{(k)}) = \prod_{u=1}^{t} (1 - f_{hazard}(T = u|T \geq u, I^{(k)})) \quad (17)$$

The most common method for estimating the hazard function is the Cox Proportional Hazards (CoxPH) model, in which f_{hazard} can be parameterized as:

$$\lambda(t|x) = \lambda_0(t)e^{\theta X} \quad (18)$$

where $\lambda_0(t)$ represents the baseline hazard function, θ represents the vector of coefficients for the covariates.

6.2. Conditional VAE

The conditional variational autoencoder (CVAE) [32] performs the variational inference with condition from the prior latent distribution. In our case, the goal is to generate target genomic features X given the pathological features Y. The variational lower bound of CVAE can be derived as follows:

$$\begin{aligned}
&\log p_\theta(X|Y) = \int_z dz q_\phi(z|X,Y) \log p_\theta(X|Y) \\
&= \int_z dz q_\phi(z|X,Y) \log p_\theta(X|Y) \\
&= \mathbb{E}_{q_\phi(z|X,Y)} \log p_\theta(X|Y) \\
&= \mathbb{E}_{q_\phi(z|X,Y)} \log \frac{p_\theta(X,Y,z)}{p_\theta(z|X,Y)p_\theta(Y)} \\
&= \mathbb{E}_{q_\phi(z|X,Y)} \log \frac{q_\phi(z|X,Y)}{p_\theta(z|X,Y)} \frac{p_\theta(X,Y,z)}{q_\phi(z|X,Y)p_\theta(Y)} \\
&= KL(q_\phi(z|X,Y)||p_\theta(z|X,Y)) + \mathbb{E}_{q_\phi(z|X,Y)} \log \frac{p_\theta(X,Y,z)}{q_\phi(z|X,Y)p_\theta(Y)} \\
&\geq \mathbb{E}_{q_\phi(z|X,Y)} \log \frac{p_\theta(X,Y,z)}{q_\phi(z|X,Y)p_\theta(Y)} \\
&= \mathbb{E}_{q_\phi(z|X,Y)} \log \frac{p_\theta(X|z,Y)p_\theta(z|Y)p_\theta(Y)}{q_\phi(z|X,Y)p_\theta(Y)} \\
&= \mathbb{E}_{q_\phi(z|X,Y)} \log p_\theta(X|z,Y) - KL(q_\phi(z|X,Y)||p_\theta(Y))
\end{aligned} \quad (19)$$

6.3. Pathological VIB Transformer

The goal of VIB [2] is to learn a new representation z_Y compressed from the original pathological information Y while conserving information about the target T. To learn the minimal sufficient representation z_Y, we can formulate it as follows (for clarity, we employ Z as an alternative for z_Y in this section):

$$\arg\max_Z I(Z,T) - \beta I(Z,Y) \quad (20)$$

where $I(\cdot)$ represents mutual information (MI). The hyperparameter $\beta \geq 0$ controls the trade-off between compression and prediction, determining the strength of the bottleneck. However, the computation of MI is intractable, VIB [2] approximates the computation of IB by using variational inference.

We firstly suppose that the joint distribution $p(Y, T, Z)$ can be calculated via:

$$p(Y,T,Z) = p(Z|Y,T)p(T|Y)p(Y) = p(Z|Y)p(T|Y)p(Y) \quad (21)$$

where we assume that $p(Z|Y,T) = p(Z|Y)$ follows the Markov chain $T \leftrightarrow Y \leftrightarrow Z$. Then, we can reformulate the

terms $I(Z,T)$ and $I(Z,Y)$ as follows:

$$\begin{aligned} I(Z,T) &= \int dt\, dz\, p(z,t) \log \frac{p(z,t)}{p(z)p(t)} \\ &= \int dt\, dz\, p(z,t) \log \frac{p(t|z)}{p(z)} \end{aligned} \tag{22}$$

$$\begin{aligned} I(Z,Y) &= \int dy\, dz\, p(z,y) \log \frac{p(z,y)}{p(z)p(y)} \\ &= \int dy\, dz\, p(z,y) \log \frac{p(z|y)}{p(z)} \end{aligned} \tag{23}$$

Then, we can derive

$$\begin{aligned} I(Z,T) &= \int dtdzp(z,t) \log \frac{p(t|z)}{p(t)} \\ &= \int dtdzp(z,t) \log \frac{q(t|z)}{p(t)} \frac{p(t|z)}{q(t|z)} \\ &= \int dtdzp(z,t) \log \frac{q(t|z)}{p(t)} + KL(p(t|z)||q(t|z)) \\ &\geq \int dtdzp(z,t) \log \frac{q(t|z)}{p(t)} \\ &= \int dtdzp(z,t) \log q(t|z) - \int dtdzp(z,t) \log p(t) \\ &= \int dtdzp(z,t) \log q(t|z) + H(t) \end{aligned} \tag{24}$$

Notice that $H(t)$ is independent in our optimization procedure and thus we can derive:

$$\begin{aligned} I(Z,T) &\geq \int dtdzp(z,t) \log q(t|z) \\ &= \int dydtdzp(y)p(z|y)p(t|y) \log q(t|z) \end{aligned} \tag{25}$$

For $I(Z,Y)$, we have

$$\begin{aligned} I(Z,Y) &= \int dydzp(z,y) \log \frac{p(y|z)}{p(y)} \\ &= \int dtdzp(z,y) \log \frac{r(z)}{p(z)} \frac{p(z|y)}{r(z)} \\ &= -KL(p(z)||r(z)) + \int dydzp(z,y) \log \frac{p(z|y)}{r(z)} \\ &\leq \int dydzp(z,y) \log \frac{p(z|y)}{r(z)} \\ &= \int dydzp(y)p(z|y) \log \frac{p(z|y)}{r(z)} \end{aligned} \tag{26}$$

By the combination of Eq. (25) and Eq. (26), we have

$$\begin{aligned} I(Z,T) - \beta I(Z,Y) &\geq \int dydtdzp(y)p(z|y)p(t|y) \log q(t|z) \\ &\quad -\beta \int dydzp(y)p(z|y) \log \frac{p(z|y)}{r(z)} \\ &\approx \int dzp(z|y) \log q(t|z) - \beta p(z|y) \log \frac{p(z|y)}{r(z)} \end{aligned} \tag{27}$$

Finally, the objective function for VIB can be denoted as:

$$\begin{aligned} \mathcal{L}_{IB} = \mathbb{E}_{z\sim p(z|y)}[-\log q(t|z)] \\ +\beta KL(p(z|y)||r(z)) \end{aligned} \tag{28}$$

where $q(t|z)$ is the variational approximation to $p(t|z)$, $r(z)$ is the variational approximation of $p(z)$ and $p(z|y)$is the posterior distribution over z.

6.4. Conditional Latent Differentiation VAE

6.4.1 Latent Differentiation VAE

In general, we assume that the N functional genomic features $x_1, x_2, ..., x_N$ are conditionally independent given a latent variable z_X. Consequently, the objective of training this VAE is to maximize the likelihood of the data $p(X) = p(x_1, x_2, , ..., x_N)$, which can be optimized using an evidence lower bound (ELBO), and the loss function can be defined as:

$$\begin{aligned} \mathcal{L}_{ELBO} = -\mathbb{E}_{q_\phi(z_X|x)}[\sum_{i=1}^{N} \log p_\theta(x_i|z_X)] \\ +\beta KL[q_\phi(z_X|X), p(z_X)] \end{aligned} \tag{29}$$

However, it is difficult to generate different functional genomic features x_i directly from the genomic posterior $p(z|x_1, x_2, ..., x_N)$ as the genomic posterior will affect the diversity of the generated genomic features [40]. To address this, we introduce the function-specific posteriors $p(z_i|X)$ by applying a latent differentiation process that transforms the genomic posterior into function-specific posteriors. Therefore, we can establish a shared latent space on all genomic features as well as refining the function-specific posteriors for each genomic category.

Specifically, we assign a unique latent variable z_i to each x_i and assume that x_i, z_i and z_X satisfy the Markov chain $x_i \leftrightarrow z_i \leftrightarrow z_X$. Due to the one-to-one correspondence, we have $p(x_i|z_X) = p(x_i, z_i|z_X)$. Then the generative model is with the following form:

$$\begin{aligned} p(x_1, z_1, x_2, z_2, ..., x_N, z_N, z_X) &= p(x_1, x_2, ..., x_N, z) \\ &= p(z_X)p(x_1|z_X)p(x_2|z_X)...p(x_N|z_X) \end{aligned} \tag{30}$$

We assume that $p(z_i|X) = \mathbb{E}_{z_X\sim p(z_X|X)}p(z_i|z_X)$, indicating that $p(z_i|X)$ can be obtained by transforming from

the genomic posterior. Since $p(z_X|X)$ is variationally approximated by $q(z_X|X)$, our function-specific posterior can be derived directly from the variational genomic posterior, avoiding the need for an independent variational approximation of $q(z_X|X)$. Therefore, we have $p(z_i|X) \approx \mathbb{E}_{z_X \sim q(z_X|X)} p(z_i|z_X)$. We model $p(z_i|z_X)$ as a process that maps $p(z_X)$ to $p(z_i)$, which can be presented as $p(z_i|z_X) = \mathcal{N}(z_i|\psi_i^{\mu}(z_X), \psi_i^{\Sigma}(z_X))$,where ψ_i serves as an MLP mapper. In practice, each function-specific posterior should also approximate prior distribution $r(z_i)$, which can be achieved by applying the Kullback-Leibler divergence.

In the process for variational inference, we sample the genomic latent variable z_X from the prior $p(z_X) \sim \mathcal{N}(0, I)$. Then, the joint distribution $p(x_i, z_i, z_X)$ can be factored as follows:

$$\begin{aligned} p(x_i, z_i, z_X) &= p(x_i|z_i, z_X)p(z_i|z_X)p(z_X) \\ &= p(x_i|z_i)p(z_i|z_X)p(z_X) \end{aligned} \tag{31}$$

Here, $p(x_i|z_i, z_X) = p(x_i|z_i)$ follows the Markov chain. Then the $\log p_\theta(x_i|z_X)$ in Eq. (6) can be reformulated as:

$$\log p(x_i|z_X) = \log p(x_i, z_i|z_X) = \log p(x_i|z_i)p(z_i|z_X) \tag{32}$$

By combining the above analysis, the loss function for LD-VAE is denoted as:

$$\begin{aligned} \mathcal{L}_{ELBO} = -\mathbb{E}_{q_\phi(z_X|X)}[\sum_{i=1}^{N} \log p_\theta(x_i|z_i)p(z_i|z_X)] \\ +\beta(\sum_{i=1}^{N} KL[q_\phi(z_i|X)||p(z_i)] + KL[q_\phi(z_X|X)||p(z_X)]) \end{aligned} \tag{33}$$

The architecture of LD-VAE. We show the detailed architecture of LD-VAE in Fig. 3(b). Specifically, we employ a transformer similar to the VIB-Trans as the encoder. The transformer encoder takes the bag of genomic feature $X = \{X_1, X_2, ..., X_N\}$ as input with two additional learnable tokens, μ_X^{token} and Σ_X^{token}, and outputs the parameters μ_X and Σ_X of the genomic posterior distribution in LD-VAE. For the reconstruction of genomic features, we set specific decoders for the genomic features with diverse biological functions. The specific decoder first uses the mapper ψ_i to generate the function-specific posterior from the genomic posterior, and then obtain the specific latent variable z_i with the re-parametrization trick to generate the genomic features x_i by the reconstruction net θ_i.

6.4.2 Joint Pathology-Genomics Distribution Learning

One critical problem for optimizing Eq. (9) is to estimate the joint posterior $q(z|X, Y)$. Following the study in [40], we assume that X,Y are conditionally independent given the genomic latent variable z, *i.e.,* $p(X, Y|z) = p(X|z)p(Y|z)$. Hence, the joint posterior can be approximated by the product of the genomics and pathology posteriors with the form:

$$\begin{aligned} p(z|X, Y) &= \frac{p(X, Y|z)p(z)}{p(X, Y)} \\ &= \frac{p(z)}{p(X, Y)} p(X|z)p(Y|z) \\ &= \frac{p(z)}{p(X, Y)} \frac{p(z|X)p(X)}{p(z)} \frac{p(z|Y)p(Y)}{p(z)} \\ &= \frac{p(z|X)p(z|Y)}{p(z)} \frac{p(X)p(Y)}{p(z)} \\ &\propto \frac{p(z|X)p(z|Y)}{p(z)} \end{aligned} \tag{34}$$

By the approximation of $p(z|X) \equiv \tilde{q}(z|X)p(z)$, $p(z|Y) \equiv \tilde{q}(z|Y)p(z)$, where $\tilde{q}(\cdot)$ is the underlying inference network, we can derive:

$$\begin{aligned} p(z|X, Y) &\propto \frac{p(z|X)p(z|Y)}{p(z)} \\ &\approx \tilde{q}(z|X)\tilde{q}(z|Y)p(z) \equiv q(z|X, Y) \end{aligned} \tag{35}$$

Finally, we can use the product-of-experts (PoE) that factorizes the joint posterior $q(z|X, Y)$ into marginal posteriors $\tilde{q}(z|X)$ and $\tilde{q}(z|Y)$.

7. Additional Experiments

7.1. Evaluation and Implementation

Evaluation. We employ 5-fold cross-validation for each dataset. To evaluate the model's performance, we calculate the concordance index (C-index) [14] and its standard deviation (std), which measure the model's ability to correctly rank pairs of individuals based on their predicted survival times. Additionally, we visualize the Kaplan-Meier (KM) [19] survival curves to illustrate the survival probability of different risk groups predicted by our model. To statistically validate the separation between risk groups, we perform the Log-rank test [27], which determines whether the survival differences between groups are statistically significant.

Implementation. For each WSI, we crop it into non-overlapping 224×224 patches at 10×magnification level. Then, a pre-trained Swin Transformer encoder (*i.e.,* CTransPath) serves as the pathological encoder φ_p, which is pre-trained using contrastive learning on over 15 million pan-cancer histopathology patches [38, 39]. For genomic data, we organizes genes into the aforementioned N=6 categories based on similar biological functional impact, which are obtained from [23]. We use the SNN [20] as the genomic encoder φ_g. Our model is implemented in Python 3.9

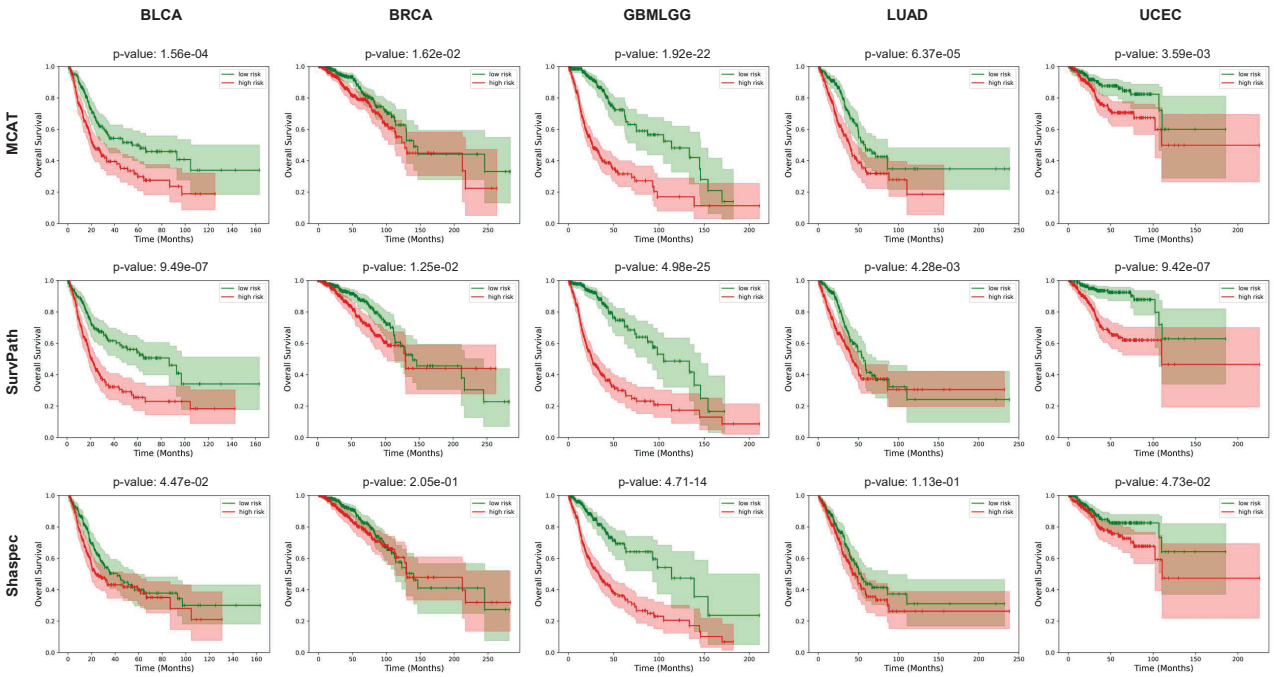

Figure 6. Kaplan-Meier Analysis of predicted high-risk (red) and low-risk (green) groups on five cancer datasets under both complete modality (top) and missing modality (bottom) scenarios. Shaded areas refer to the confidence intervals.

Table 4. Comparisons of C-Index (mean ± std) with different methods for the representation of WSIs over five datasets. For each method, results are reported under both complete modality and missing modality scenarios.

Model	Missing	BLCA	BRCA	GBMLGG	LUAD	UCEC	Overall
ABMIL		0.651 ± 0.035	0.685 ± 0.020	0.833 ± 0.058	0.646 ± 0.048	0.691 ± 0.053	0.701
	✓	0.616 ± 0.024	0.619 ± 0.019	0.820 ± 0.040	0.623 ± 0.035	0.647 ± 0.037	0.665
CLAM-SM		0.651 ± 0.035	0.685 ± 0.020	0.833 ± 0.058	0.646 ± 0.048	0.691 ± 0.053	0.701
	✓	0.616 ± 0.024	0.619 ± 0.019	0.820 ± 0.040	0.623 ± 0.035	0.647 ± 0.037	0.665
CLAM-MB		0.651 ± 0.035	0.685 ± 0.020	0.833 ± 0.058	0.646 ± 0.048	0.691 ± 0.053	0.701
	✓	0.616 ± 0.024	0.619 ± 0.019	0.820 ± 0.040	0.623 ± 0.035	0.647 ± 0.037	0.665
TransMIL		0.651 ± 0.035	0.685 ± 0.020	0.833 ± 0.058	0.646 ± 0.048	0.691 ± 0.053	0.701
	✓	0.616 ± 0.024	0.619 ± 0.019	0.820 ± 0.040	0.623 ± 0.035	0.647 ± 0.037	0.665
VIB-Trans		0.686 ± 0.035	0.680 ± 0.030	0.849 ± 0.017	0.676 ±0.015	0.703 ± 0.069	0.719
	✓	0.649 ± 0.040	0.641 ± 0.012	0.821 ± 0.021	0.628 ± 0.008	0.681 ± 0.044	0.684

with Pytorch library and trained with four NVIDIA 3090 GPUs. During training, we set the hyperparameter α to 0.1. The hyperparameter β for Kullback-Leibler divergence is annealed using a cosine schedule, gradually increasing to 1, thereby forming a valid lower bound on the evidence. We adopt Adam optimizer with the initial learning rate of 2×10^{-4} and weight decay of 1×10^{-5}. Following the setting of [5], we use the batch size of 1 for WSIs with 32 gradient accumulation steps, and all experiments are trained for 30 epoches. We train our methods on the complete data and test the performance on both the complete modality and missing modality (*i.e.,* genomic data).

7.2. Additional Results

Additional Results for Patient Stratification. We present additional results for patient stratification using competing methods, including those designed for integrating multi-modal data (*e.g.,* MCAT, SurvPath) and handling missing genomic data (*e.g.,* Shaspace), as shown in Fig. 6. Compared to these methods, our approach (shown in Fig. 4) achieves clearer separation between low-risk and high-risk patients across all datasets. In the Logrank test, our method can still consistently yield a lower P-value. These results highlight the robustness and effectiveness of our approach in accurately distinguishing patient risk groups.

Table 5. Ablation study assessing C-index (mean ± std) over five datasets. For each variant, results are reported under both complete modality and missing modality scenarios.

Variants	Missing	BLCA	BRCA	GBMLGG	LUAD	UCEC	Overall
w/o μ^{token} and Σ^{token}		0.675 ± 0.063	0.674 ± 0.033	0.839 ± 0.035	0.668 ± 0.039	0.682 ± 0.081	0.708
	✓	0.641 ± 0.040	0.644 ± 0.012	0.811 ± 0.027	0.629 ± 0.022	0.643 ± 0.036	0.674
Ours		0.686 ± 0.035	0.680 ± 0.030	0.849 ± 0.017	0.676 ±0.015	0.703 ± 0.069	0.719
	✓	0.649 ± 0.040	0.641 ± 0.012	0.821 ± 0.021	0.628 ± 0.008	0.681 ± 0.044	0.684

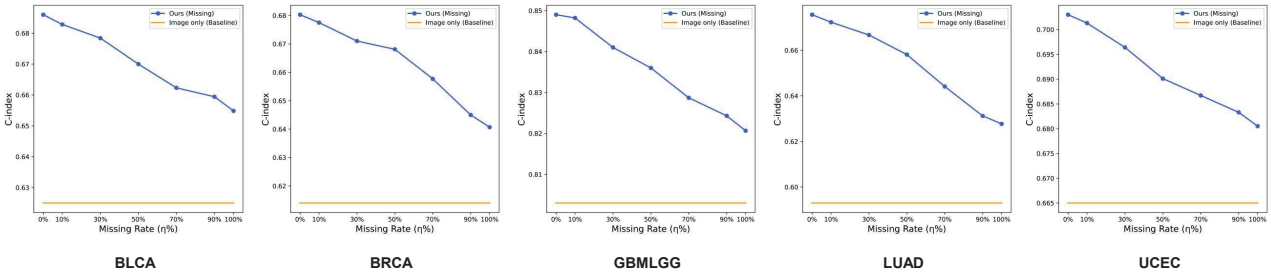

Figure 7. The performance of our method under different settings of missing rate η.

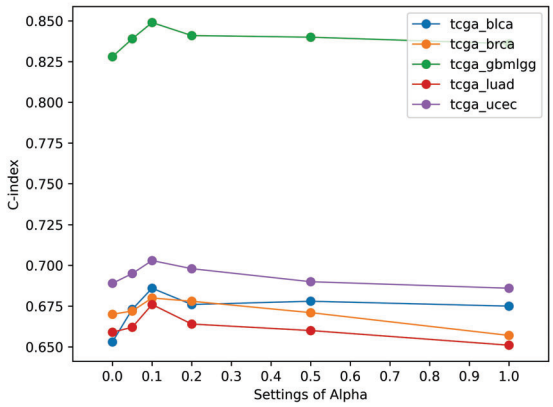

Figure 8. The effect of hyperparameter α over five cancer datasets.

Representation of WSIs. We conduct additional experiments with different methods to represent WSIs in place of VIB-Trans, with the results summarized in Tab. 4. We can observe that VIB-Trans consistently outperforms competing methods across all datasets, demonstrating its effectiveness of representing WSIs.

Settings of missing rate for genomic data. We conduct further experiments to analyze the robustness of our proposed method against different missing rates for genomic data, and the results are shwon in Fig. 7. We observe that the increase of missing rate η comes with inferior prediction results. However, even at high missing rates of 90% or even 100%, our method still outperforms the best unimodal approach basing on pathology images, indicating the effectiveness of our approach.

Settings of hyperparameters for Loss Function. We conduct ablation studies to evaluate the impact of the hyperparameter α in Eq. (13). As shown in Fig. 8, model performance peaks when α is 0.1, hence we set 0.1 as the optimal value of α.

Impact of $\mu^{token}, \sigma^{token}$. We introduce the μ^{token} and Σ^{token}in Trans-VIB to better learn the latent posterior of VIB by modeling the interaction between pairwise patches. For ablation, we use two linear layers following the transformer encoder to obtain the μ and Σ. As shown in Tab. 5, the inclusion of μ^{token} and Σ^{token} achieves superior performance, indicating the effectiveness of this design.

7.3. More Comparisons with State-of-the-Arts

We also conduct additional experiments with pathological features extracted by a ResNet50 encoder pre-trained on ImageNet, and report the results with comparisons to SOTA methods reported in Tabs. 6 and 7. The results in Tab. 6 demonstrate that our method consistently achieves the best overall performance across both unimodal and multimodal approaches. In the missing modality setting, results in Tab. 7 show that our method can effectively handle missing genomic data and outperform the comparison methods.

7.4. More Visualizations

We provide more visualizations that compare the co-attention weights calculated from the ground truth and generated genomic feature in Figs. 9 to 13.

Table 6. Comparisons of C-index (mean ± std) with SOTA methods over five cancer datasets by using ResNet50 encoder. g. and h. refer to genomic modality and histological modality, respectively. The best results and the second-best results are highlighted in **bold** and in underline.

Model	Modality	BLCA (N=373)	BRCA (N=957)	GBMLGG (N=571)	LUAD (N=452)	UCEC (N=480)	Overall
MLP	g.	0.613 ± 0.019	0.587 ± 0.033	0.809 ± 0.029	0.617 ± 0.026	0.657 ± 0.036	0.657
SNN	g.	0.619 ± 0.023	0.596 ± 0.027	0.805 ± 0.030	0.625 ± 0.019	0.651 ± 0.018	0.659
SNNTrans	g.	0.627 ± 0.019	0.618 ± 0.018	0.816 ± 0.037	0.631 ± 0.023	0.641 ± 0.026	0.667
ABMIL	h.	0.594 ± 0.033	0.601 ± 0.033	0.779 ± 0.035	0.579 ± 0.070	0.637 ± 0.024	0.638
CLAM-SB	h.	0.594 ± 0.047	0.595 ± 0.028	0.787 ± 0.036	0.580 ± 0.053	0.648 ± 0.032	0.641
CLAM-MB	h.	0.598 ± 0.030	0.600 ± 0.017	0.790 ± 0.031	0.582 ± 0.077	0.657 ± 0.038	0.645
TransMIL	h.	0.605 ± 0.054	0.604 ± 0.054	0.793 ± 0.028	0.590 ± 0.057	0.649 ± 0.053	0.648
Porpoise	g.+h.	0.646 ± 0.038	0.652 ± 0.022	0.819 ± 0.033	0.649 ± 0.030	0.665 ± 0.043	0.685
MCAT	g.+h.	0.645 ± 0.031	0.648 ± 0.011	0.826 ± 0.033	0.651 ± 0.043	0.659 ± 0.062	0.690
MOTCat	g.+h.	0.649 ± 0.016	0.646 ± 0.055	0.829 ± 0.039	0.654 ± 0.031	0.651 ± 0.053	0.687
CMTA	g.+h.	0.653 ± 0.035	0.656 ± 0.045	**0.837 ± 0.028**	0.657 ± 0.029	0.660 ± 0.035	0.693
SurvPath	g.+h.	0.651±0.028	0.667 ± 0.053	0.833 ±0.043	0.660 ± 0.015	0.674 ± 0.051	0.697
PIBD	g.+h.	0.611 ± 0.012	0.606 ± 0.020	0.783 ± 0.056	0.621 ± 0.013	0.632 ± 0.038	0.651
Ours	g.+h.	**0.678 ± 0.048**	**0.668±0.042**	0.835 ± 0.037	**0.664 ± 0.035**	**0.680 ± 0.049**	**0.703**

Table 7. Comparisons of C-index (mean ± std) with methods addressing missing modality over five cancer datasets by using ResNet50 encoder. The best results and the second-best results are highlighted in **bold** and in underline.

Model	BLCA (N=373)	BRCA (N=957)	GBMLGG (N=571)	LUAD (N=452)	UCEC (N=480)	Overall
VAE	0.588 ± 0.016	0.618 ± 0.028	0.788 ± 0.019	0.588 ± 0.044	0.636 ± 0.034	0.643
GAN	0.585 ± 0.011	0.611 ± 0.026	0.779 ± 0.029	0.599 ± 0.051	0.637 ± 0.044	0.642
MVAE	0.588 ± 0.028	0.612 ± 0.029	0.774 ± 0.015	0.601 ± 0.026	0.636 ± 0.024	0.642
SMIL	0.610 ± 0.020	0.615 ± 0.015	0.775 ± 0.021	0.599 ± 0.024	0.647 ± 0.024	0.649
ShaSpec	0.615 ± 0.017	0.618 ± 0.028	0.791 ± 0.011	0.611 ± 0.040	0.656 ± 0.036	**0.658**
Transformer	0.613 ± 0.033	0.620 ± 0.022	0.797 ± 0.022	0.602 ± 0.039	0.652 ± 0.013	0.657
Ours	**0.637±0.028**	**0.634 ± 0.033**	**0.806 ± 0.026**	**0.634 ± 0.038**	**0.661 ± 0.029**	**0.672**

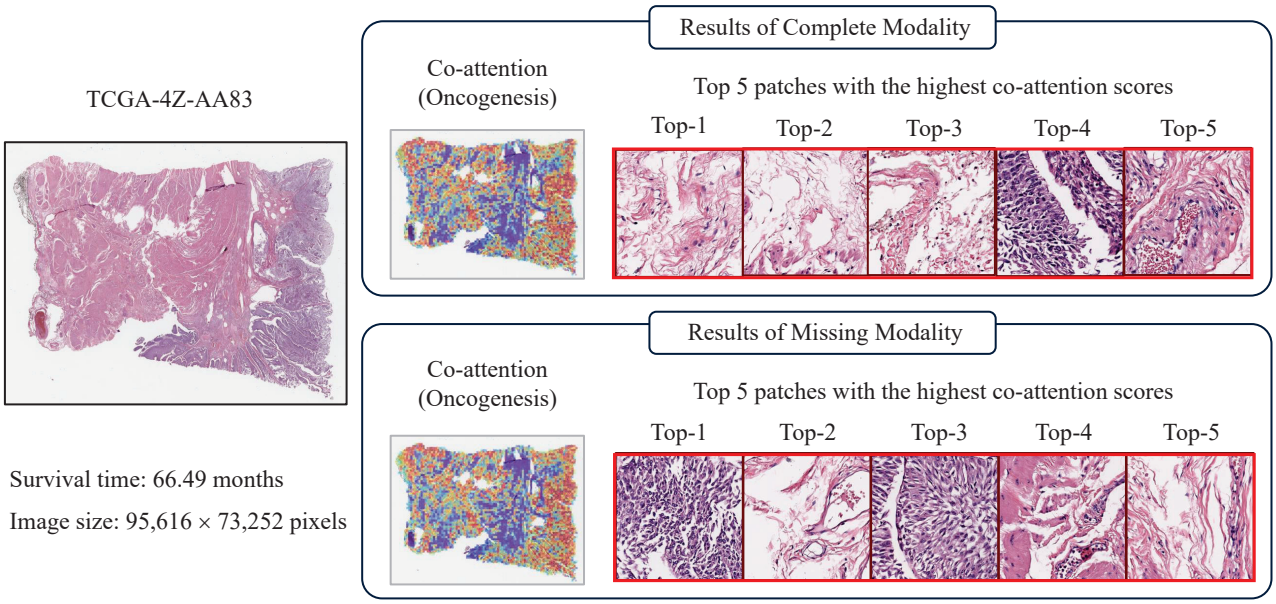

Figure 9. Comparison of the co-attention weights calculated from the genuine (top) and generated (bottom) genomic features.

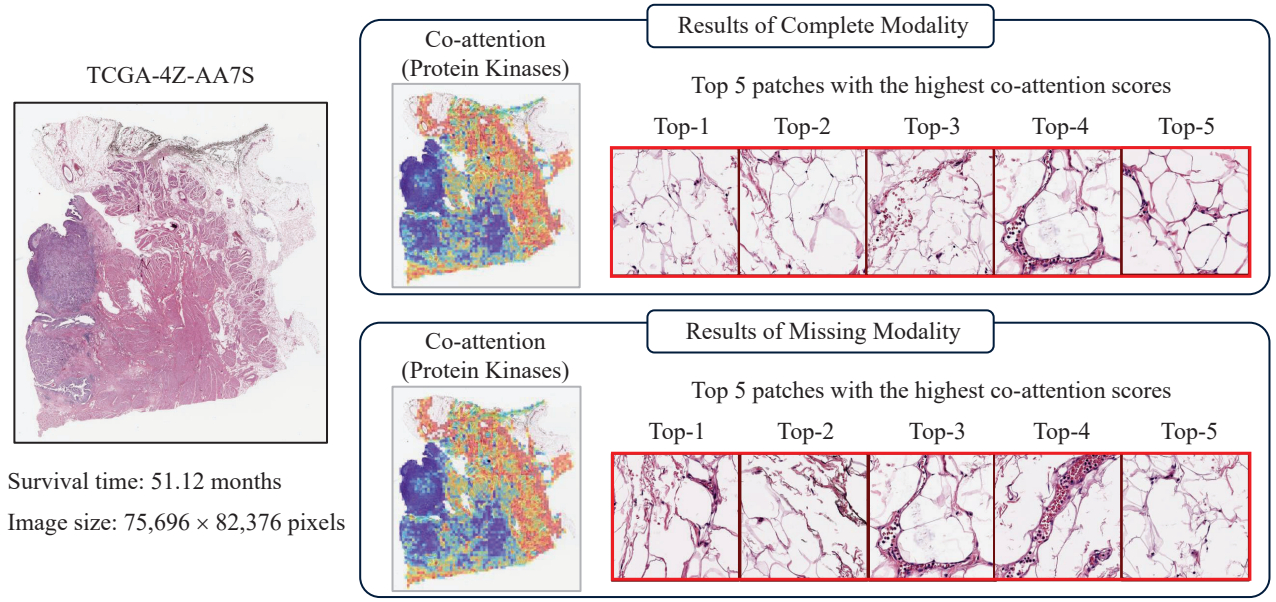

Figure 10. Comparison of the co-attention weights calculated from the genuine (top) and generated (bottom) genomic features.

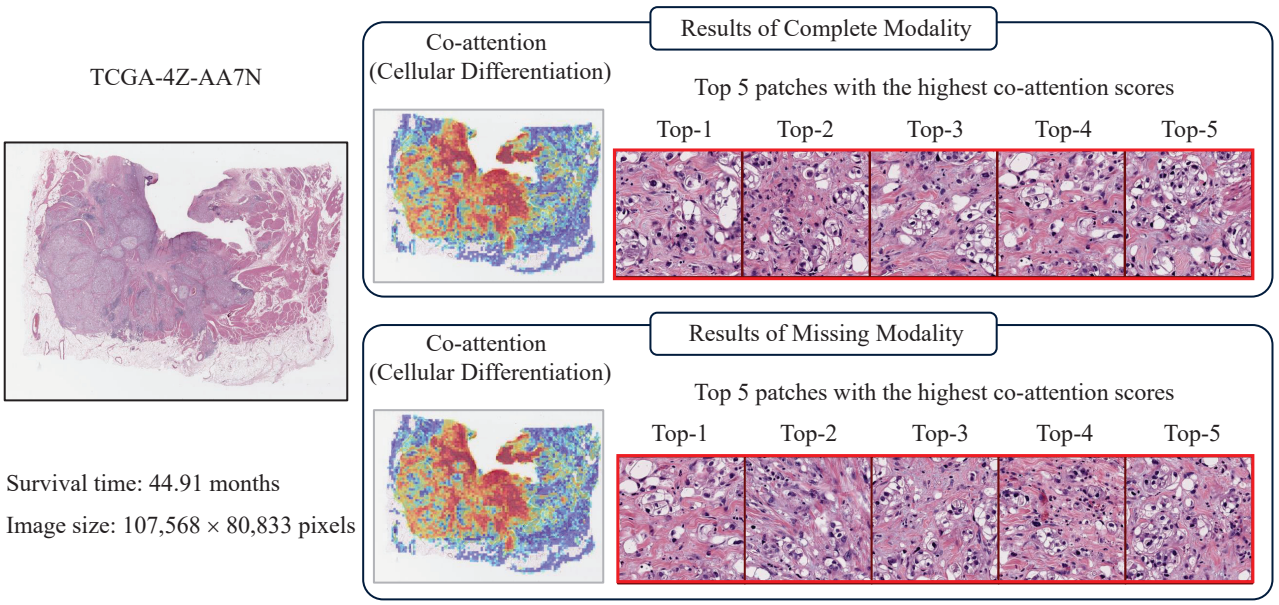

Figure 11. Comparison of the co-attention weights calculated from the genuine (top) and generated (bottom) genomic features.

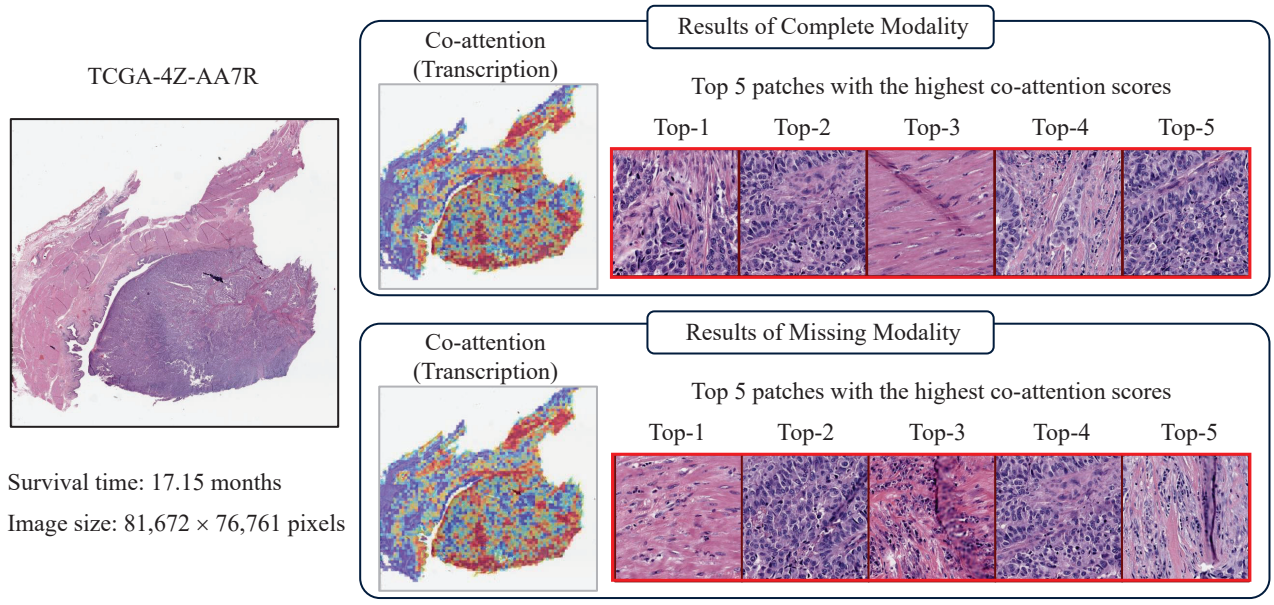

Figure 12. Comparison of the co-attention weights calculated from the genuine (top) and generated (bottom) genomic features.

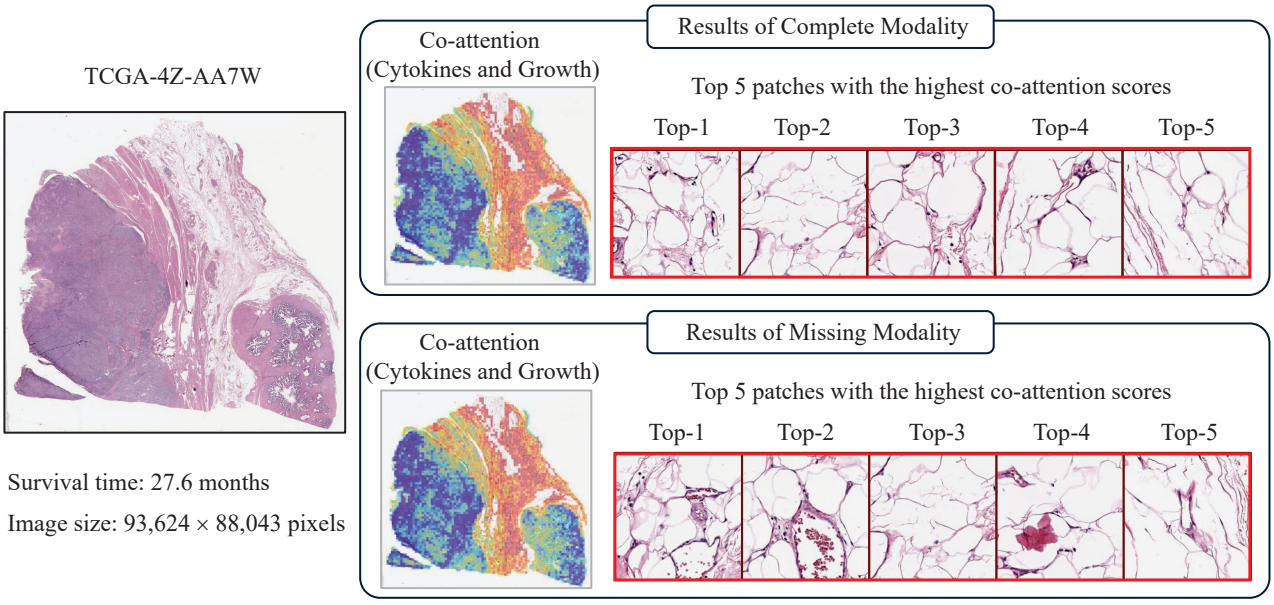

Figure 13. Comparison of the co-attention weights calculated from the genuine (top) and generated (bottom) genomic features.