Concept Paper

Bayesian Multi-View Graph Convolutional Network (BMGCN) for Integrative Multi-Omics Analysis with Survival Outcomes and Zero-Inflated Data

Akhilesh Kaushal *

Posted Date: 27 October 2025

doi: 10.20944/preprints202510.1979.v1

Keywords: multi-omics; Bayesian

 doi:10.20944/preprints202510.1979.v1

Concept Paper

Bayesian Multi-View Graph Convolutional Network (BMGCN) for Integrative Multi-Omics Analysis with Survival Outcomes and Zero-Inflated Data

Akhilesh Kaushal

Department of Pediatric Hematology, University of Arkansas for Medical Sciences, USA; akhileshkaushal@gmail.com

Abstract

Modern biomedical research generates vast, multi-modal datasets (multi-omics) from the same patient cohorts, offering an unprecedented opportunity to understand complex diseases. However, integrating these heterogeneous data views to predict clinical outcomes like patient survival presents significant statistical challenges. These challenges include data heterogeneity, high dimensionality, inherent zero-inflation due to technical dropouts or biological absence, and the need to incorporate prior biological knowledge. We propose the Bayesian Multi-view Graph Convolutional Network (BMGCN), a deep generative framework designed to address these challenges. BMGCN factorizes the data into shared and view-specific latent representations, enabling both data integration and the identification of view-specific signals. It employs graph-convolutional encoders to integrate prior biological network knowledge, a zero-inflated likelihood to accurately model sparse omics data, and a spike-and-slab prior for Bayesian view selection to identify modalities most relevant to the outcome. Finally, a semi-parametric Cox proportional hazards module allows the model to handle right-censored survival data directly. We detail the full generative model, derive the variational inference objective, and outline a comprehensive validation strategy. BMGCN provides a powerful, interpretable, and flexible framework for integrative multi-omics analysis.

Keywords: multi-omics; Bayesian

1. Introduction

The ability to measure multiple molecular layers—such as gene expression (RNA-seq), DNA methylation, microRNA, and proteomics—from the same subjects has revolutionized systems biology and precision medicine. Each of these "omics" views provides a unique window into the biological state of a system. The central hypothesis of multi-omics integration is that a unified analysis of these views can reveal emergent biological insights and yield more accurate predictive models than any single view alone [1].

However, the integrative analysis of such data is fraught with challenges. First, the data are heterogeneous, with different views having unique statistical properties, dimensionalities, and levels of noise. Second, many omics datasets, particularly those from single-cell and sequencing-based assays, are characterized by a high degree of sparsity and zero-inflation, where an excess of zero values complicates modeling [2]. These zeros can represent true biological absence or technical artifacts (e.g., "dropouts"), and distinguishing between them is crucial. Third, predicting clinical endpoints, such as patient survival, requires specialized statistical models that can properly handle right-censored data, where the event of interest is not observed for all subjects [3]. Finally, ignoring the rich web of prior biological knowledge, such as protein-protein interaction networks or gene regulatory pathways, is a missed opportunity to guide the model toward more robust and interpretable solutions.

To address these multifaceted challenges, we introduce the Bayesian Multi-view Graph Convolutional Network (BMGCN). BMGCN is a deep generative framework that provides a principled

 doi:10.20944/preprints202510.1979.v1

solution for integrating multi-omics data with censored survival outcomes. The design of BMGCN is guided by the following core objectives:

1. **Latent Abstraction:** To learn a low-dimensional representation of the data by factorizing the signal into latent variables that are shared across all omics views ($\mathbf{Z}_i$) and variables that are specific to each view ($\mathbf{Z}_i^{(k)}$). This captures both common and view-specific biological processes.
2. **View Selection:** To automatically determine the relevance of each omics view for predicting the survival outcome. We achieve this through a Bayesian spike-and-slab prior on the view-specific latents (γ_k), which can effectively "turn off" uninformative views, enhancing interpretability [4].
3. **Principled Sparsity Modeling:** To explicitly model the excess zeros common in omics data using a zero-inflated mixture likelihood [2]. This distinguishes between true biological absence (a point mass at zero) and low-level biological signal (a continuous component).
4. **Censored Outcome Modeling:** To directly model time-to-event data through an integrated Cox proportional hazards module that operates on the learned latent representations [3].
5. **Integration of Biological Priors:** To leverage prior knowledge in the form of biological networks ($\mathbf{A}^{(k)}$) by using Graph Convolutional Network (GCN) encoders, which regularize the model and encourage biologically meaningful latent representations [5].

This paper details the statistical foundation of BMGCN. We first present the notation and model overview, followed by a detailed description of the full generative process. We then derive the variational inference framework and the Evidence Lower Bound (ELBO) objective function. Finally, we describe the architecture of the graph-based encoders and outline a rigorous strategy for model validation and diagnostics.

2. Methods

2.1. Notation and Core Entities

Let our dataset consist of N subjects and K omics views. The core entities of BMGCN are defined as follows:

Observed Data:

- $\mathbf{X}_i^{(k)} \in \mathbb{R}^{p_k}$: The feature vector for subject i in view k, where p_k is the number of features in that view.
- $\mathbf{A}^{(k)} \in \{0,1\}^{p_k \times p_k}$: An optional adjacency matrix for view k, representing prior biological knowledge (e.g., a protein-protein interaction network).
- (T_i, Δ_i): The observed time for subject i, where $T_i \in \mathbb{R}^+$ is the survival or censoring time and $\Delta_i \in \{0,1\}$ is the event indicator ($\Delta_i = 1$ if the event was observed, $\Delta_i = 0$ if censored).

Latent Variables:

- $\mathbf{Z}_i \in \mathbb{R}^L$: A shared latent vector for subject i, capturing variation common to all views.
- $\mathbf{Z}_i^{(k)} \in \mathbb{R}^{L_k}$: A view-specific latent vector for subject i in view k.
- $\gamma_k \in \{0,1\}$: A binary variable indicating whether view k is active in predicting survival.
- $\pi_k \in (0,1)$: The prior inclusion probability for view k.

2.2. The BMGCN Generative Model

BMGCN defines a full generative process for the observed data, which factorizes into several key steps. For each subject $i = 1, \ldots, N$:

1. **Sample View Inclusion Probabilities and Indicators:** For each view $k = 1, \ldots, K$, we first draw a prior inclusion probability from a Beta distribution and then sample the binary indicator variable:

$$\pi_k \sim \text{Beta}(\alpha_0, \beta_0) \tag{1}$$

$$\gamma_k \mid \pi_k \sim \text{Bernoulli}(\pi_k) \tag{2}$$

 doi:10.20944/preprints202510.1979.v1

2. **Sample Latent Variables:** A shared latent vector is drawn from a standard normal prior. The view-specific latent vectors are drawn from a spike-and-slab prior, conditioned on the inclusion variable γ_k. If $\gamma_k = 0$, the latent vector is fixed at zero (the "spike"); otherwise, it is drawn from a standard normal (the "slab").

$$\mathbf{Z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L) \tag{3}$$

$$p(\mathbf{Z}_i^{(k)} \mid \gamma_k) = (1 - \gamma_k)\delta_{\mathbf{0}}(\mathbf{Z}_i^{(k)}) + \gamma_k \mathcal{N}(\mathbf{Z}_i^{(k)} \mid \mathbf{0}, \mathbf{I}_{L_k}) \tag{4}$$

where $\delta_{\mathbf{0}}$ is the Dirac delta function at zero.

3. **Generate Omics Features via Decoders:** The observed features for each view are generated from the combined shared and view-specific latent variables. A decoder network $f_\psi^{(k)}$ maps the concatenated latent vector $[\mathbf{Z}_i, \mathbf{Z}_i^{(k)}]$ to the parameters of a zero-inflated Gaussian distribution.

$$(\boldsymbol{\mu}_i^{(k)}, \boldsymbol{\rho}_i^{(k)}) = f_\psi^{(k)}([\mathbf{Z}_i, \mathbf{Z}_i^{(k)}]) \tag{5}$$

$$p(X_{ij}^{(k)} \mid \dots) = \rho_{ij}^{(k)} \delta_0(X_{ij}^{(k)}) + (1 - \rho_{ij}^{(k)}) \mathcal{N}(X_{ij}^{(k)} \mid \mu_{ij}^{(k)}, \sigma_k^2) \tag{6}$$

Here, $\boldsymbol{\mu}_i^{(k)}$ is the mean vector, $\boldsymbol{\rho}_i^{(k)}$ is the vector of zero-inflation probabilities (passed through a sigmoid function to ensure they are in $(0, 1)$), and σ_k^2 is a view-specific variance parameter.

4. **Generate Survival Outcome:** A latent risk score η_i for each subject is computed by a survival head network g_θ, which takes the shared latent and the gated view-specific latents as input. The observed survival data (T_i, Δ_i) are assumed to follow a Cox proportional hazards model conditioned on this risk score.

$$\eta_i = g_\theta([\mathbf{Z}_i, \gamma_1 \mathbf{Z}_i^{(1)}, \dots, \gamma_K \mathbf{Z}_i^{(K)}]) \tag{7}$$

$$(T_i, \Delta_i) \sim \text{Cox}(\eta_i) \tag{8}$$

The likelihood for the Cox model is given by the partial log-likelihood:

$$l(\boldsymbol{\eta}) = \sum_{i=1}^{N} \Delta_i \left(\eta_i - \log \sum_{j \in R_i} e^{\eta_j} \right) \tag{9}$$

where $R_i = \{j : T_j \geq T_i\}$ is the set of subjects at risk at time T_i. This semi-parametric form is powerful because it does not require specifying a baseline hazard function $h_0(t)$. When multiple events occur at the same time (ties), approximations such as Breslow's or Efron's can be used. The full joint distribution over all observed and latent variables is given by:

$$p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = \left[\prod_{i=1}^{N} p(T_i, \Delta_i \mid \eta_i) \prod_{k=1}^{K} p(\mathbf{X}_i^{(k)} \mid \mathbf{Z}_i, \mathbf{Z}_i^{(k)}) p(\mathbf{Z}_i^{(k)} \mid \gamma_k) p(\mathbf{Z}_i) \right] \times \left[\prod_{k=1}^{K} p(\gamma_k \mid \pi_k) p(\pi_k) \right] \tag{10}$$

2.3. Variational Inference and the ELBO

The posterior distribution over the latent variables, $p(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma} \mid \mathbf{X}, \mathbf{T}, \boldsymbol{\Delta})$, is intractable due to the non-linearities in the model and the Cox likelihood. We therefore resort to amortized variational inference (VI) [6,7]. We introduce a tractable variational distribution $q(\cdot)$ to approximate the true posterior and maximize the Evidence Lower Bound (ELBO), $\mathcal{L}$, with respect to the variational parameters.

 doi:10.20944/preprints202510.1979.v1

We use a mean-field variational family that factorizes across subjects and latent variables:

$$q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \gamma) = \prod_{i=1}^{N} q(\mathbf{Z}_i) \prod_{k=1}^{K} q(\mathbf{Z}_i^{(k)}) q(\gamma_k) \tag{11}$$

where each factor is parameterized by an encoder network:

$$q(\mathbf{Z}_i \mid \mathbf{X}_i) = \mathcal{N}(\boldsymbol{\mu}_i(\mathbf{X}_i), \mathrm{diag}(\boldsymbol{\sigma}_i^2(\mathbf{X}_i))) \tag{12}$$

$$q(\mathbf{Z}_i^{(k)} \mid \mathbf{X}_i^{(k)}, \mathbf{A}^{(k)}) = \mathcal{N}(\boldsymbol{\mu}_i^{(k)}(\mathbf{X}_i^{(k)}, \mathbf{A}^{(k)}), \mathrm{diag}((\boldsymbol{\sigma}_i^{(k)})^2(\mathbf{X}_i^{(k)}, \mathbf{A}^{(k)}))) \tag{13}$$

$$q(\gamma_k) = \mathrm{Bernoulli}(\tau_k) \tag{14}$$

where τ_k is the posterior inclusion probability for view k.

The ELBO is defined as:

$$\mathcal{L} = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \gamma)] - \mathbb{E}_q[\log q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \gamma)] \tag{15}$$

This objective can be decomposed into several intuitive terms:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{surv}} - \mathcal{L}_{\text{KL}} \tag{16}$$

where:

1. **Reconstruction Term ($\mathcal{L}_{\text{recon}}$):** The expected log-likelihood of the omics data.

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{E}_q[\log p(\mathbf{X}_i^{(k)} \mid \mathbf{Z}_i, \mathbf{Z}_i^{(k)})] \tag{17}$$

   The piecewise nature of the zero-inflated log-likelihood makes this term directly computable without explicit sampling of discrete zero-inflation masks.

2. **Survival Term ($\mathcal{L}_{\text{surv}}$):** The expected Cox partial log-likelihood.

$$\mathcal{L}_{\text{surv}} = \mathbb{E}_q \left[\sum_{i=1}^{N} \Delta_i \left(\eta_i - \log \sum_{j \in R_i} e^{\eta_j} \right) \right] \tag{18}$$

3. **KL Divergence Regularizer ($\mathcal{L}_{\text{KL}}$):** A sum of Kullback-Leibler (KL) divergences that penalize deviations of the variational posterior from the prior.

$$\mathcal{L}_{\text{KL}} = \sum_{i=1}^{N} \mathrm{KL}(q(\mathbf{Z}_i) \,\|\, p(\mathbf{Z}_i)) \tag{19}$$

$$+ \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{q(\gamma_k)}[\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, p(\mathbf{Z}_i^{(k)} \mid \gamma_k))] \tag{20}$$

$$+ \sum_{k=1}^{K} \mathrm{KL}(q(\gamma_k) \,\|\, p(\gamma_k)) \tag{21}$$

   The KL terms for the Gaussian latents have a closed-form solution. The term involving the discrete γ_k is approximated as $\tau_k \cdot \mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I}))$, effectively weighting the KL penalty by the posterior probability that the view is included.

2.4. Graph-Convolutional Encoders

To incorporate prior biological network information ($\mathbf{A}^{(k)}$), the encoder networks that parameterize $q(\mathbf{Z}_i^{(k)})$ are implemented as Graph Convolutional Networks (GCNs) [5]. This allows the model

 doi:10.20944/preprints202510.1979.v1

to learn feature representations that are regularized by known biological interactions, improving robustness and interpretability.

For each view k, we first normalize the adjacency matrix to ensure stable learning:

$$\hat{\mathbf{A}}^{(k)} = (\tilde{\mathbf{D}}^{(k)})^{-1/2}\tilde{\mathbf{A}}^{(k)}(\tilde{\mathbf{D}}^{(k)})^{-1/2} \tag{22}$$

where $\tilde{\mathbf{A}}^{(k)} = \mathbf{A}^{(k)} + \mathbf{I}$ (adding self-loops) and $\tilde{\mathbf{D}}^{(k)}$ is the diagonal degree matrix of $\tilde{\mathbf{A}}^{(k)}$.

The GCN takes the full data matrix for view k, $\mathbf{X}^{(k)} \in \mathbb{R}^{N\times p_k}$, as input features for the nodes (e.g., genes) of the graph. The GCN then learns feature embeddings by propagating and transforming information between nodes in the graph. The propagation rule for layer l is:

$$\mathbf{H}^{(l+1)} = \text{ReLU}(\hat{\mathbf{A}}^{(k)}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \tag{23}$$

where $\mathbf{H}^{(0)} = (\mathbf{X}^{(k)})^\top \in \mathbb{R}^{p_k\times N}$, $\mathbf{W}^{(l)}$ is a learnable weight matrix, and $\mathbf{H}^{(l)} \in \mathbb{R}^{p_k\times d_l}$ is the feature embedding matrix at layer l. After L layers, we obtain a final feature embedding matrix $\mathbf{H}^{(L)} \in \mathbb{R}^{p_k\times d_L}$.

To obtain sample-specific parameters for $q(\mathbf{Z}_i^{(k)})$, we apply a readout function. Specifically, the final feature embeddings are used to parameterize a sample-specific network that maps the input data to the latent space parameters:

$$\mathbf{h}_i^{(k)} = \text{FFN}_{\text{enc}}(\mathbf{X}_i^{(k)}, \mathbf{H}^{(L)}) \tag{24}$$

This sample-specific embedding $\mathbf{h}_i^{(k)}$ is then mapped to the variational parameters:

$$\boldsymbol{\mu}_i^{(k)} = \mathbf{W}_\mu\mathbf{h}_i^{(k)} \tag{25}$$

$$\log\boldsymbol{\sigma}_i^{(k)} = \mathbf{W}_\sigma\mathbf{h}_i^{(k)} \tag{26}$$

2.5. Optimization and Implementation

The ELBO is optimized with respect to the parameters of the encoders, decoders, and survival head using stochastic gradient ascent. We employ the reparameterization trick for Gaussian latents to obtain low-variance gradient estimates [6]. For the discrete view-selection variables γ_k, we use a hard-concrete relaxation with a straight-through gradient estimator during training [8,9].

Training is performed in mini-batches. For the Cox survival term, which is defined over the full cohort, we use mini-batch approximations, such as constructing risk sets only from subjects within the current batch or using a memory bank. While this introduces a small bias, it is essential for scalability.

Key implementation details for ensuring stable training include:

- Using the log-sum-exp trick for stable computation of the Cox likelihood denominator.
- Applying KL annealing to gradually introduce the KL regularization term, preventing posterior collapse [10].
- Using modern optimizers like AdamW and employing learning rate schedulers [11,12].
- Initializing the view-selection logits (τ_k) to favor inclusion initially to promote stable learning.

3. Validation and Diagnostics Strategy

To ensure BMGCN is robust, generalizable, and interpretable, we propose a multi-faceted validation strategy.

- **Simulation Studies:** We will generate synthetic multi-view data with known ground-truth latent structures, pre-defined informative vs. uninformative views, and known survival dependencies. This will allow us to quantitatively assess the model's ability to recover the true latent variables (cosine similarity), correctly perform view selection (accuracy of τ_k vs. ground truth), and accurately predict survival (Concordance Index).

 doi:10.20944/preprints202510.1979.v1

- **Cross-Validation:** On real-world datasets, we will use a k-fold cross-validation scheme, stratified by event status, to evaluate predictive performance on held-out data. The primary metric will be the Concordance Index (C-index), which measures the fraction of concordant pairs of subjects. A secondary metric will be the integrated Brier score, which assesses the calibration of survival probability predictions over time.
- **Posterior Predictive Checks (PPCs):** To assess goodness-of-fit, we will sample from the posterior predictive distribution. We will compare the distributions of simulated data (e.g., zero frequencies, means, variances) against the observed data to detect model misspecification.
- **Cox Diagnostics:** We will examine Schoenfeld residuals to test the proportional hazards assumption of the Cox model, a critical assumption for the validity of the survival module.
- **Ablation Studies:** To quantify the contribution of each model component, we will perform ablation studies by systematically removing key features: (1) the graph structure (setting $\mathbf{A}^{(k)} = \mathbf{I}$), (2) the zero-inflation module ($\boldsymbol{\rho} = \mathbf{0}$), and (3) the spike-and-slab selectors (forcing $\gamma_k = 1$ for all views).
- **Latent Space Analysis:** We will visualize the learned shared latent space $\mathbf{Z}_i$ using techniques like UMAP to check if it clusters subjects by known phenotypes, treatment groups, or survival outcomes, providing a qualitative check on its biological relevance [13].

4. Discussion

BMGCN offers a comprehensive and principled framework for one of the most pressing problems in computational biology: the integrative analysis of multi-omics data for clinical outcome prediction. Its primary strengths lie in its ability to simultaneously address multiple, often co-occurring, challenges.

The Bayesian framework provides not only point estimates but also uncertainty quantification. The probabilistic view selection mechanism offers a path to identifying which data modalities are most informative for a given prediction task, a crucial step for biomarker discovery and for designing future cost-effective studies.

However, the model is not without limitations. The mean-field approximation used for variational inference may not capture complex posterior dependencies between latent variables. The Cox model component relies on the proportional hazards assumption, which may not hold in all biological contexts; diagnostics are essential. Computationally, BMGCN is intensive, particularly with a large number of views or features, though scalability is addressed via mini-batching.

Future work could extend BMGCN in several directions. More flexible variational families, such as normalizing flows, could be explored to improve posterior approximation. The model could be extended to handle other data types (e.g., categorical or count data) and other clinical endpoints. Furthermore, incorporating mechanisms to explicitly handle missing views for certain subjects would greatly enhance its applicability to real-world clinical datasets.

5. Conclusions

We have presented BMGCN, a Bayesian deep generative model for the integrative analysis of multi-omics data with censored survival outcomes. By combining shared and view-specific latent variables, graph-convolutional encoders, a zero-inflated likelihood, and a Cox survival head within a unified probabilistic framework, BMGCN is tailored to the unique challenges of modern biomedical data. Its design prioritizes flexibility, interpretability, and statistical rigor, providing a powerful new tool for uncovering the complex molecular underpinnings of disease and for improving patient outcome prediction.

Appendix A. Supplementary Mathematical Details: Full Derivations

This appendix provides a complete and rigorous mathematical derivation of the core components of the Bayesian Multi-view Graph Convolutional Network (BMGCN). We derive the Evidence Lower

 doi:10.20944/preprints202510.1979.v1

Bound (ELBO), its decomposition into constituent terms, the gradients for optimization, and the closed-form solutions for key quantities.

Appendix A.1. Derivation of the Evidence Lower Bound (ELBO)

The goal of variational inference is to approximate the true, intractable posterior distribution $p(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi} \mid \mathbf{X}, \mathbf{T}, \boldsymbol{\Delta})$ with a tractable variational distribution $q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi})$. We assume a mean-field factorization for q as specified in the main text:

$$q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = q(\mathbf{Z})q(\{\mathbf{Z}^{(k)}\})q(\boldsymbol{\gamma})q(\boldsymbol{\pi}) = \left[\prod_{i=1}^{N} q(\mathbf{Z}_i)\right]\left[\prod_{i=1}^{N}\prod_{k=1}^{K} q(\mathbf{Z}_i^{(k)})\right]\left[\prod_{k=1}^{K} q(\boldsymbol{\gamma}_k)\right]\left[\prod_{k=1}^{K} q(\boldsymbol{\pi}_k)\right] \tag{A1}$$

The quality of this approximation is measured by the Kullback-Leibler (KL) divergence between q and p:

$$\mathrm{KL}(q \,\|\, p) = \mathbb{E}_q[\log q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi})] - \mathbb{E}_q[\log p(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi} \mid \mathbf{X}, \mathbf{T}, \boldsymbol{\Delta})] \tag{A2}$$

By Bayes' theorem, the log-posterior is:

$$\log p(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi} \mid \mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}) = \log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi}) - \log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}) \tag{A3}$$

Substituting this into the KL divergence:

$$\mathrm{KL}(q \,\|\, p) = \mathbb{E}_q[\log q] - \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi})] + \log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}) \tag{A4}$$

Rearranging terms, we get:

$$\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi})] - \mathbb{E}_q[\log q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi})] + \mathrm{KL}(q \,\|\, p) \tag{A5}$$

Since $\mathrm{KL}(q \,\|\, p) \geq 0$, the first two terms on the right-hand side form a lower bound on the log-evidence (marginal likelihood) $\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta})$. This is the Evidence Lower Bound (ELBO), denoted by $\mathcal{L}$:

$$\mathcal{L} = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi})] - \mathbb{E}_q[\log q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi})] \tag{A6}$$

This is Equation (15) from the main text. The ELBO is maximized with respect to the parameters of q to find the best approximation to the true posterior.

Appendix A.2. Decomposition of the ELBO

We now decompose the ELBO $\mathcal{L}$ into its constituent parts: the reconstruction term ($\mathcal{L}_{\text{recon}}$), the survival term ($\mathcal{L}_{\text{surv}}$), and the KL divergence regularizer ($\mathcal{L}_{\text{KL}}$).

Starting from the joint distribution defined in Equation (10):

$$p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma}, \boldsymbol{\pi}) = \left[\prod_{i=1}^{N} p(T_i, \Delta_i \mid \eta_i) \prod_{k=1}^{K} p(\mathbf{X}_i^{(k)} \mid \mathbf{Z}_i, \mathbf{Z}_i^{(k)}) p(\mathbf{Z}_i^{(k)} \mid \boldsymbol{\gamma}_k) p(\mathbf{Z}_i)\right] \times \left[\prod_{k=1}^{K} p(\boldsymbol{\gamma}_k \mid \boldsymbol{\pi}_k) p(\boldsymbol{\pi}_k)\right] \tag{A7}$$

 doi:10.20944/preprints202510.1979.v1

Taking the expectation under q:

$$\begin{aligned}\mathbb{E}_q[\log p(\mathbf{X},\mathbf{T},\boldsymbol{\Delta},\mathbf{Z},\{\mathbf{Z}^{(k)}\},\boldsymbol{\gamma},\boldsymbol{\pi})] =& \mathbb{E}_q\left[\sum_{i=1}^{N}\log p(T_i,\Delta_i \mid \eta_i)\right] + \mathbb{E}_q\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log p(\mathbf{X}_i^{(k)} \mid \mathbf{Z}_i,\mathbf{Z}_i^{(k)})\right] \\ &+ \mathbb{E}_q\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log p(\mathbf{Z}_i^{(k)} \mid \gamma_k)\right] + \mathbb{E}_q\left[\sum_{i=1}^{N}\log p(\mathbf{Z}_i)\right] \\ &+ \mathbb{E}_q\left[\sum_{k=1}^{K}\log p(\gamma_k \mid \pi_k)\right] + \mathbb{E}_q\left[\sum_{k=1}^{K}\log p(\pi_k)\right] \end{aligned} \tag{A8}$$

The second term in the ELBO is the entropy of the variational distribution:

$$\begin{aligned}\mathbb{E}_q[\log q(\mathbf{Z},\{\mathbf{Z}^{(k)}\},\boldsymbol{\gamma},\boldsymbol{\pi})] =& \mathbb{E}_q\left[\sum_{i=1}^{N}\log q(\mathbf{Z}_i)\right] + \mathbb{E}_q\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log q(\mathbf{Z}_i^{(k)})\right] \\ &+ \mathbb{E}_q\left[\sum_{k=1}^{K}\log q(\gamma_k)\right] + \mathbb{E}_q\left[\sum_{k=1}^{K}\log q(\pi_k)\right] \end{aligned} \tag{A9}$$

Substituting these into the ELBO $\mathcal{L}$:

$$\begin{aligned}\mathcal{L} =& \underbrace{\mathbb{E}_q\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log p(\mathbf{X}_i^{(k)} \mid \mathbf{Z}_i,\mathbf{Z}_i^{(k)})\right]}_{\mathcal{L}_{\text{recon}}} + \underbrace{\mathbb{E}_q\left[\sum_{i=1}^{N}\log p(T_i,\Delta_i \mid \eta_i)\right]}_{\mathcal{L}_{\text{surv}}} \\ &- \underbrace{\left(\mathbb{E}_q\left[\sum_{i=1}^{N}\log q(\mathbf{Z}_i)\right] - \mathbb{E}_q\left[\sum_{i=1}^{N}\log p(\mathbf{Z}_i)\right]\right)}_{\text{KL for } \mathbf{Z}_i} \\ &- \underbrace{\left(\mathbb{E}_q\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log q(\mathbf{Z}_i^{(k)})\right] - \mathbb{E}_q\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log p(\mathbf{Z}_i^{(k)} \mid \gamma_k)\right]\right)}_{\text{KL for } \mathbf{Z}_i^{(k)}} \\ &- \underbrace{\left(\mathbb{E}_q\left[\sum_{k=1}^{K}\log q(\gamma_k)\right] - \mathbb{E}_q\left[\sum_{k=1}^{K}\log p(\gamma_k \mid \pi_k)\right]\right)}_{\text{KL for } \gamma_k} \\ &- \underbrace{\left(\mathbb{E}_q\left[\sum_{k=1}^{K}\log q(\pi_k)\right] - \mathbb{E}_q\left[\sum_{k=1}^{K}\log p(\pi_k)\right]\right)}_{\text{KL for } \pi_k} \end{aligned} \tag{A10}$$

This can be grouped as follows:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_q\left[\sum_{i=1}^{N}\sum_{k=1}^{K}\log p(\mathbf{X}_i^{(k)} \mid \mathbf{Z}_i,\mathbf{Z}_i^{(k)})\right] \quad \text{(Reconstruction Term)} \tag{A11}$$

$$\mathcal{L}_{\text{surv}} = \mathbb{E}_q\left[\sum_{i=1}^{N}\log p(T_i,\Delta_i \mid \eta_i)\right] = \mathbb{E}_q\left[\sum_{i=1}^{N}\Delta_i\left(\eta_i - \log\sum_{j\in R_i}e^{\eta_j}\right)\right] \quad \text{(Survival Term)} \tag{A12}$$

$$\begin{aligned}\mathcal{L}_{\text{KL}} =& \sum_{i=1}^{N}\text{KL}(q(\mathbf{Z}_i)\,\|\,p(\mathbf{Z}_i)) + \sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}_q\left[\text{KL}(q(\mathbf{Z}_i^{(k)})\,\|\,p(\mathbf{Z}_i^{(k)} \mid \gamma_k))\right] \\ &+ \sum_{k=1}^{K}\text{KL}(q(\gamma_k)\,\|\,p(\gamma_k \mid \pi_k)) + \sum_{k=1}^{K}\text{KL}(q(\pi_k)\,\|\,p(\pi_k)) \end{aligned} \tag{A13}$$

 doi:10.20944/preprints202510.1979.v1

This confirms the decomposition $\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{surv}} - \mathcal{L}_{\text{KL}}$ from Equation (16). For simplicity in the main text, the priors on γ_k and π_k are often absorbed or their KL terms are handled implicitly, leading to the simplified $\mathcal{L}_{\text{KL}}$ shown in Equations (19-21).

Appendix A.3. Zero-Inflated Gaussian Log-Likelihood and Gradients

The probability density function (PDF) for the zero-inflated Gaussian (ZIG) distribution for a single feature j in view k for subject i is:

$$p(X_{ij}^{(k)} \mid \mu_{ij}^{(k)}, \sigma_k^2, \rho_{ij}^{(k)}) = \rho_{ij}^{(k)} \cdot \delta_0(X_{ij}^{(k)}) + (1 - \rho_{ij}^{(k)}) \cdot \mathcal{N}(X_{ij}^{(k)} \mid \mu_{ij}^{(k)}, \sigma_k^2) \tag{A14}$$

where $\delta_0(x)$ is the Dirac delta function, which is infinite at $x = 0$ and zero elsewhere, with the property that $\int \delta_0(x)dx = 1$.

The **log-likelihood** for an observation x is piecewise, as it must handle the discrete mass at zero and the continuous density elsewhere:

$$\log p(x \mid \mu, \sigma^2, \rho) = \begin{cases} \log\left(\rho + (1-\rho) \cdot \mathcal{N}(0 \mid \mu, \sigma^2)\right) & \text{if } x = 0 \\ \log(1-\rho) + \log \mathcal{N}(x \mid \mu, \sigma^2) & \text{if } x \neq 0 \end{cases} \tag{A15}$$

To optimize the model, we need the **gradients** of this log-likelihood with respect to its parameters μ, σ^2, and ρ.

Case 1: $x = 0$

Let $C = \rho + (1-\rho) \cdot \mathcal{N}(0 \mid \mu, \sigma^2)$.

$$\begin{aligned} \frac{\partial}{\partial \mu}[\log C] &= \frac{1}{C} \cdot (1-\rho) \cdot \frac{\partial}{\partial \mu}[\mathcal{N}(0 \mid \mu, \sigma^2)] \\ &= \frac{1}{C} \cdot (1-\rho) \cdot \mathcal{N}(0 \mid \mu, \sigma^2) \cdot \left(\frac{(0-\mu)}{\sigma^2}\right) \end{aligned} \tag{A16}$$

$$\begin{aligned} \frac{\partial}{\partial \sigma^2}[\log C] &= \frac{1}{C} \cdot (1-\rho) \cdot \frac{\partial}{\partial \sigma^2}[\mathcal{N}(0 \mid \mu, \sigma^2)] \\ &= \frac{1}{C} \cdot (1-\rho) \cdot \mathcal{N}(0 \mid \mu, \sigma^2) \cdot \left[\frac{(0-\mu)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right] \end{aligned} \tag{A17}$$

$$\frac{\partial}{\partial \rho}[\log C] = \frac{1}{C} \cdot \left[1 - \mathcal{N}(0 \mid \mu, \sigma^2)\right] \tag{A18}$$

Case 2: $x \neq 0$

$$\frac{\partial}{\partial \mu}[\log(1-\rho) + \log \mathcal{N}(x \mid \mu, \sigma^2)] = \frac{\partial}{\partial \mu}[\log \mathcal{N}(x \mid \mu, \sigma^2)] = \frac{(x-\mu)}{\sigma^2} \tag{A19}$$

$$\frac{\partial}{\partial \sigma^2}[\log(1-\rho) + \log \mathcal{N}(x \mid \mu, \sigma^2)] = \frac{\partial}{\partial \sigma^2}[\log \mathcal{N}(x \mid \mu, \sigma^2)] = \left[\frac{(x-\mu)^2}{2\sigma^4} - \frac{1}{2\sigma^2}\right] \tag{A20}$$

$$\frac{\partial}{\partial \rho}[\log(1-\rho) + \log \mathcal{N}(x \mid \mu, \sigma^2)] = -\frac{1}{(1-\rho)} \tag{A21}$$

These gradients are used in the backpropagation algorithm to update the parameters of the decoder network $f_\psi^{(k)}$.

Appendix A.4. Gradient of the Cox Partial Log-Likelihood

The Cox partial log-likelihood for the entire dataset is:

$$l(\boldsymbol{\eta}) = \sum_{i=1}^{N} \Delta_i \left(\eta_i - \log \sum_{j \in R_i} e^{\eta_j}\right) \tag{A22}$$

 doi:10.20944/preprints202510.1979.v1

where $R_i = \{j : T_j \geq T_i\}$ is the risk set for subject i.

We derive the gradient of $l(\boldsymbol{\eta})$ with respect to the risk score η_m for a specific subject m.

The gradient $\partial l / \partial \eta_m$ receives contributions from two places in the summation:

1. When $i = m$, if subject m experienced the event ($\Delta_m = 1$), then η_m appears directly in the first term.
2. For any subject i for whom subject m is in the risk set (i.e., $m \in R_i$), η_m appears in the log-sum-exp term $\log(\sum_{j \in R_i} e^{\eta_j})$.

Therefore:

$$\begin{aligned} \frac{\partial l}{\partial \eta_m} &= \Delta_m \cdot 1 - \sum_{i: m \in R_i} \Delta_i \cdot \left[\frac{1}{\sum_{j \in R_i} e^{\eta_j}} \right] \cdot e^{\eta_m} \\ &= \Delta_m - \sum_{i: m \in R_i} \Delta_i \cdot \left[\frac{e^{\eta_m}}{\sum_{j \in R_i} e^{\eta_j}} \right] \end{aligned} \tag{A23}$$

This gradient is used to update the parameters of the survival head network g_θ.

Appendix A.5. KL Divergence for Gaussian Distributions

We derive the closed-form solution for the KL divergence between two multivariate Gaussian distributions, which is used for the latent variables $\mathbf{Z}_i$ and $\mathbf{Z}_i^{(k)}$.

Let $q(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ and $p(\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$. The KL divergence is:

$$\mathrm{KL}(q \parallel p) = \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z})] \tag{A24}$$

The expectation of the log of a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is:

$$\mathbb{E}_q[\log \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})] = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\mathbb{E}_q[(\mathbf{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})] \tag{A25}$$

For $q(\mathbf{z})$, this is:

$$\begin{aligned} \mathbb{E}_q[\log q(\mathbf{z})] &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_q| - \frac{1}{2}\mathbb{E}_q[(\mathbf{z}-\boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1}(\mathbf{z}-\boldsymbol{\mu}_q)] \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_q| - \frac{1}{2}\mathrm{trace}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q) \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_q| - \frac{d}{2} \end{aligned} \tag{A26}$$

because $\mathbb{E}_q[(\mathbf{z}-\boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_q^{-1}(\mathbf{z}-\boldsymbol{\mu}_q)] = \mathrm{trace}(\boldsymbol{\Sigma}_q^{-1}\mathbb{E}_q[(\mathbf{z}-\boldsymbol{\mu}_q)(\mathbf{z}-\boldsymbol{\mu}_q)^\top]) = \mathrm{trace}(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_q) = \mathrm{trace}(\mathbf{I}) = d$.

For $p(\mathbf{z})$, assuming a standard normal prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ (so $\boldsymbol{\mu}_p = \mathbf{0}, \boldsymbol{\Sigma}_p = \mathbf{I}$), we have:

$$\begin{aligned} \mathbb{E}_q[\log p(\mathbf{z})] &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{I}| - \frac{1}{2}\mathbb{E}_q[\mathbf{z}^\top \mathbf{z}] \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_q\left[\sum_{l=1}^{d} z_l^2\right] \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{l=1}^{d}\mathbb{E}_q[z_l^2] \end{aligned} \tag{A27}$$

Since $\mathbb{E}_q[z_l^2] = \mathrm{Var}_q(z_l) + (\mathbb{E}_q[z_l])^2 = \sigma_{q,l}^2 + \mu_{q,l}^2$, we get:

$$\mathbb{E}_q[\log p(\mathbf{z})] = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{l=1}^{d}(\mu_{q,l}^2 + \sigma_{q,l}^2) \tag{A28}$$

 doi:10.20944/preprints202510.1979.v1

Subtracting Equation (A28) from Equation (A26):

$$\begin{aligned}\mathrm{KL}(q \,\|\, p) &= \left[-\frac{1}{2}\log|\boldsymbol{\Sigma}_q| - \frac{d}{2}\right] - \left[-\frac{1}{2}\sum_{l=1}^{d}(\mu_{q,l}^2 + \sigma_{q,l}^2)\right] \\ &= \frac{1}{2}\sum_{l=1}^{d}(\mu_{q,l}^2 + \sigma_{q,l}^2 - \log(\sigma_{q,l}^2) - 1) \end{aligned} \tag{A29}$$

where we have used the fact that for a diagonal covariance matrix $\boldsymbol{\Sigma}_q = \mathrm{diag}(\sigma_{q,1}^2, \ldots, \sigma_{q,d}^2)$, the determinant $|\boldsymbol{\Sigma}_q| = \prod_{l=1}^{d}\sigma_{q,l}^2$, and thus $\log|\boldsymbol{\Sigma}_q| = \sum_{l=1}^{d}\log(\sigma_{q,l}^2)$.

Appendix A.6. KL Divergence for the Spike-and-Slab Prior

For the view-specific latent $\mathbf{Z}_i^{(k)}$, the prior is a spike-and-slab: $p(\mathbf{Z}_i^{(k)} \mid \gamma_k) = (1-\gamma_k)\delta_{\mathbf{0}}(\mathbf{Z}_i^{(k)}) + \gamma_k\mathcal{N}(\mathbf{Z}_i^{(k)} \mid \mathbf{0}, \mathbf{I})$

The variational posterior is a Gaussian: $q(\mathbf{Z}_i^{(k)}) = \mathcal{N}(\boldsymbol{\mu}_q^{(k)}, \mathrm{diag}((\boldsymbol{\sigma}_q^{(k)})^2))$

The KL divergence $\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, p(\mathbf{Z}_i^{(k)} \mid \gamma_k))$ is not standard because the prior is a mixture. We handle this by taking the expectation over the discrete variable γ_k.

$$\begin{aligned}\mathbb{E}_{q(\gamma_k)}\Big[\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, p(\mathbf{Z}_i^{(k)} \mid \gamma_k))\Big] &= \mathbb{E}_{q(\gamma_k)}\left[\int q(\mathbf{Z}_i^{(k)})\log\left(\frac{q(\mathbf{Z}_i^{(k)})}{p(\mathbf{Z}_i^{(k)} \mid \gamma_k)}\right)d\mathbf{Z}_i^{(k)}\right] \\ &= q(\gamma_k = 0)\cdot\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, \delta_{\mathbf{0}}) + q(\gamma_k = 1)\cdot\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I}))\end{aligned} \tag{A30}$$

The term $\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, \delta_{\mathbf{0}})$ is problematic because the delta function is not a density with respect to the Lebesgue measure. In practice, as noted in the main text, when $\gamma_k = 0$, the prior forces $\mathbf{Z}_i^{(k)} = \mathbf{0}$. Therefore, the model approximates this by only applying the KL penalty against the Gaussian slab when the view is included. This leads to the practical approximation:

$$\mathbb{E}_{q(\gamma_k)}\Big[\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, p(\mathbf{Z}_i^{(k)} \mid \gamma_k))\Big] \approx \tau_k \cdot \mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})) \tag{A31}$$

where $\tau_k = q(\gamma_k = 1)$. This approximation is used in Equation (20).

Appendix A.7. Derivation of the Variational Objective for the Spike-and-Slab Prior

The exact KL divergence for the view-specific latent, marginalized over γ_k, is:

$$\begin{aligned}\mathrm{KL}\Big(q(\mathbf{Z}_i^{(k)}) \,\|\, p(\mathbf{Z}_i^{(k)})\Big) &= \int q(\mathbf{Z}_i^{(k)})\log\frac{q(\mathbf{Z}_i^{(k)})}{p(\mathbf{Z}_i^{(k)})}d\mathbf{Z}_i^{(k)} \\ &= \int q(\mathbf{Z}_i^{(k)})\log q(\mathbf{Z}_i^{(k)})d\mathbf{Z}_i^{(k)} - \int q(\mathbf{Z}_i^{(k)})\log\left[\int p(\mathbf{Z}_i^{(k)} \mid \gamma_k)p(\gamma_k)d\gamma_k\right]d\mathbf{Z}_i^{(k)} \\ &= -\mathcal{H}\Big[q(\mathbf{Z}_i^{(k)})\Big] - \mathbb{E}_{q(\mathbf{Z}_i^{(k)})}\Big[\log\Big((1-\pi_k)\delta_{\mathbf{0}}(\mathbf{Z}_i^{(k)}) + \pi_k\mathcal{N}(\mathbf{Z}_i^{(k)} \mid \mathbf{0}, \mathbf{I})\Big)\Big]\end{aligned} \tag{A32}$$

where $\mathcal{H}\Big[q(\mathbf{Z}_i^{(k)})\Big]$ is the entropy of the variational distribution.

The intractability arises from the log of a sum inside the expectation. The common variational approach is to introduce an auxiliary variational distribution $r(\gamma_k)$ for the discrete variable and derive a lower bound. However, in the context of the overall ELBO, this can become very complex.

The approximation used in the paper, $\mathbb{E}_{q(\gamma_k)}\Big[\mathrm{KL}(q(\mathbf{Z}_i^{(k)}) \,\|\, p(\mathbf{Z}_i^{(k)} \mid \gamma_k))\Big]$, is itself a variational approximation that treats γ_k as independent. This is known as the "fully-factorized" or "mean-field" assumption across the discrete and continuous latents. The approximation in Equation (A31) (using $\tau_k \cdot \mathrm{KL}(q \,\|\, \mathcal{N})$) is a further simplification that ignores the KL divergence to the spike (delta function)

 doi:10.20944/preprints202510.1979.v1

because it is infinite for any non-degenerate q. This is justified by the model's structure: when $\gamma_k = 0$, the survival head g_θ receives $\mathbf{0}$, so the model is incentivized to set $q(\mathbf{Z}_i^{(k)})$ to $\delta_{\mathbf{0}}$ to minimize the reconstruction loss, making the infinite KL penalty moot. The τ_k weighting then acts as a regularizer only when the view is likely to be included.

Appendix A.8. Derivation of the Optimal Variational Distribution for π_k

The appendix mentions a variational distribution $q(\pi_k)$ but does not derive its optimal form. For a conjugate model, we can find a closed-form update.

From the joint distribution in Equation (A7), the conditional posterior for π_k (before variational approximation) is proportional to:

$$\begin{aligned} p(\pi_k \mid \gamma_k) &\propto p(\gamma_k \mid \pi_k) p(\pi_k) \\ &\propto \pi_k^{\gamma_k} (1-\pi_k)^{1-\gamma_k} \cdot \pi_k^{\alpha_0 - 1} (1-\pi_k)^{\beta_0 - 1} \\ &\propto \pi_k^{\alpha_0 + \gamma_k - 1} (1-\pi_k)^{\beta_0 + 1 - \gamma_k - 1} \end{aligned} \tag{A33}$$

This shows that the conditional posterior for π_k is $\text{Beta}(\alpha_0 + \gamma_k, \beta_0 + 1 - \gamma_k)$.

In variational inference, the optimal $q^\star(\pi_k)$ that maximizes the ELBO is the one that minimizes the KL divergence to the true conditional posterior. Under the mean-field assumption, the optimal $q^\star(\pi_k)$ is:

$$\begin{aligned} q^\star(\pi_k) &\propto \exp\Big(\mathbb{E}_{q(\gamma_k)}[\log p(\pi_k \mid \gamma_k)]\Big) \\ &\propto \exp\Big(\mathbb{E}_{q(\gamma_k)}[\log \text{Beta}(\pi_k \mid \alpha_0 + \gamma_k, \beta_0 + 1 - \gamma_k)]\Big) \end{aligned} \tag{A34}$$

Since the expectation of γ_k under q is τ_k, we can substitute to find:

$$\begin{aligned} q^\star(\pi_k) &\propto \pi_k^{\alpha_0 + \tau_k - 1} (1-\pi_k)^{\beta_0 + 1 - \tau_k - 1} \\ &= \text{Beta}(\pi_k \mid \alpha_0 + \tau_k, \beta_0 + 1 - \tau_k) \end{aligned} \tag{A35}$$

Therefore, the optimal variational distribution for π_k is $q(\pi_k) = \text{Beta}(\alpha_0 + \tau_k, \beta_0 + 1 - \tau_k)$. The corresponding KL divergence term $\text{KL}(q(\pi_k) \,\|\, p(\pi_k))$ in Equation (A13) can then be computed in closed form using the KL divergence between two Beta distributions.

Appendix A.9. Graph Convolutional Layer: Signal Processing Interpretation

The propagation rule $\mathbf{H}^{(l+1)} = \text{ReLU}(\hat{\mathbf{A}}^{(k)} \mathbf{H}^{(l)} \mathbf{W}^{(l)})$ can be analyzed from a signal processing perspective on graphs.

The normalized adjacency matrix $\hat{\mathbf{A}}^{(k)}$ can be decomposed via its eigenvectors: $\hat{\mathbf{A}}^{(k)} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues and $\mathbf{U}$ is the matrix of eigenvectors. This is the graph Fourier transform.

A graph convolution in the spatial domain (like the GCN layer) is equivalent to a multiplication in the spectral domain. The operation $\hat{\mathbf{A}}^{(k)} \mathbf{H}^{(l)}$ is a form of *low-pass filter*. It smooths the feature representation of each node by averaging it with the features of its neighbors. The eigenvalues in $\mathbf{\Lambda}$ are in the range $[-1, 1]$, and the largest eigenvalues correspond to the lowest frequencies (most global, smoothest signals on the graph).

The weight matrix $\mathbf{W}^{(l)}$ then acts as a learnable filter that can amplify or attenuate different frequency components. The ReLU non-linearity introduces the capacity to learn more complex, non-linear feature interactions.

This mathematical interpretation justifies why GCNs are effective for biological data: they promote smoothness of the latent representations with respect to the prior biological network, meaning that genes/proteins that are known to interact will have similar representations in the latent space, leading to more biologically plausible and robust models.

 doi:10.20944/preprints202510.1979.v1

Appendix A.10. Hard-Concrete Relaxation for γ_k

The main text mentions using a "hard-concrete relaxation with a straight-through gradient estimator" for γ_k. Here, we provide the mathematical formulation of this trick.

The binary variable $\gamma_k \in \{0, 1\}$ is replaced during training with a continuous relaxation $\tilde{\gamma}_k \in (0, 1)$ sampled from a Hard-Concrete distribution. The sampling process is:

$$\begin{aligned} u &\sim \text{Uniform}(0, 1) \\ s &= \sigma\left(\frac{1}{\beta}(\log u - \log(1-u) + \log \alpha)\right) \\ \tilde{s} &= s \cdot (\gamma_{\max} - \gamma_{\min}) + \gamma_{\min} \\ \tilde{\gamma}_k &= \min(1, \max(0, \tilde{s})) \end{aligned} \tag{A36}$$

where α is the location parameter (often derived from the logit of τ_k), β is a temperature parameter, and $[\gamma_{\min}, \gamma_{\max}]$ is a small interval (e.g., $[-0.1, 1.1]$) to stretch the output.

During the forward pass, $\tilde{\gamma}_k$ is used. For the backward pass (gradient computation), the "straight-through" estimator is used: the gradient is computed as if the operation was an identity function. That is, if $y = \text{hard\_concrete}(x)$, then $\partial y / \partial x = 1$.

This allows gradients to flow through the discrete sampling process, enabling end-to-end training. As training progresses, the temperature β is annealed to zero, causing the Hard-Concrete distribution to collapse to a Bernoulli distribution, effectively discretizing $\tilde{\gamma}_k$ back to γ_k.

Appendix A.11. Derivation of Gradients for Model Parameters

The appendix derives the ELBO with respect to the variational parameters. It is also crucial to derive the gradients with respect to the *model parameters* ψ (decoder) and θ (survival head).

The ELBO is:

$$\mathcal{L} = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma})] - \mathbb{E}_q[\log q(\mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma})] \tag{A37}$$

The second term (the entropy of q) does not depend on the model parameters ψ, θ. Therefore, the gradient with respect to ψ and θ comes only from the first term, the expected joint log-likelihood.

$$\nabla_{\psi,\theta} \mathcal{L} = \nabla_{\psi,\theta} \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{T}, \boldsymbol{\Delta}, \mathbf{Z}, \{\mathbf{Z}^{(k)}\}, \boldsymbol{\gamma})] \tag{A38}$$

This expectation can be broken down as shown in Equation (A8). The gradients for the reconstruction term $\mathcal{L}_{\text{recon}}$ with respect to ψ are derived in Appendix A.3. The gradients for the survival term $\mathcal{L}_{\text{surv}}$ with respect to θ are derived in Appendix A.4.

The key point is that these gradients can be estimated using Monte Carlo sampling from the variational distribution q, combined with the reparameterization trick for the Gaussian latents and the straight-through estimator for the discrete latents, making the entire model trainable via standard stochastic gradient descent.

Appendix A.12. Reparameterization Trick for Gaussian Latents

To obtain low-variance, unbiased gradient estimates for the Gaussian latent variables, we use the reparameterization trick.

For a latent variable $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, instead of sampling $\mathbf{z}$ directly, we sample an auxiliary noise variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and compute $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}$, where $\boldsymbol{\Sigma}^{1/2}$ is the matrix square root. For a diagonal covariance $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$, this simplifies to element-wise multiplication: $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\odot$ denotes the Hadamard (element-wise) product.

 doi:10.20944/preprints202510.1979.v1

The gradient of an expectation $\mathbb{E}_{q(\mathbf{z})}[f(\mathbf{z})]$ with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ can then be computed as:

$$\frac{\partial}{\partial \boldsymbol{\mu}} \mathbb{E}_{q(\mathbf{z})}[f(\mathbf{z})] = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \cdot \frac{\partial \mathbf{z}}{\partial \boldsymbol{\mu}}\right] = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}}\right] \tag{A39}$$

$$\frac{\partial}{\partial \boldsymbol{\sigma}} \mathbb{E}_{q(\mathbf{z})}[f(\mathbf{z})] = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \cdot \frac{\partial \mathbf{z}}{\partial \boldsymbol{\sigma}}\right] = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \cdot \boldsymbol{\epsilon}\right] \tag{A40}$$

This allows gradients to flow through the sampling process, enabling efficient optimization via stochastic gradient descent.

References

1. Ritchie, M.D.; Holzinger, E.R.; Li, R.; Pendergrass, S.A.; Kim, D. Methods of integrating data to uncover genotype–phenotype interactions. *Nature Reviews Genetics* **2015**, *16*, 85–97.
2. Lambert, D. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* **1992**, *34*, 1–14.
3. Cox, D.R. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* **1972**, *34*, 187–202.
4. George, E.I.; McCulloch, R.E. Variable selection via Gibbs sampling. *Journal of the American statistical association* **1993**, *88*, 881–889.
5. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.
6. Kingma, D.P.; Welling, M. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114* **2013**.
7. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082* **2014**.
8. Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with Gumbel-Softmax. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.
9. Maddison, C.J.; Mnih, A.; Teh, Y.W. The concrete distribution: A continuous relaxation of discrete random variables. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.
10. Bowman, S.R.; Vilnis, L.; Vinyals, O.; Dai, A.M.; Jozefowicz, R.; Bengio, S. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* **2015**.
11. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
12. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* **2017**.
13. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* **2018**.