BMC Bioinformatics Research Article

CalibratedGraph: uncertainty-aware graph learning for robust cancer subtype classification from incomplete multi-omics profiles

Maya R. Sen[1*], Daniel K. Okafor[2], and Elena V. Hart[1,3]

[1]Department of Computational Biology, Northbridge Institute of Technology, Northbridge, USA

[2]Center for Statistical Genomics, Meridian University, Lagos, Nigeria

[3]Institute for Precision Oncology, St. Catherine Research Hospital, Northbridge, USA

[*] Correspondence: maya.sen@example.org

ORCID: 0000-0002-1825-0097

Template note. This is a complete, compilable demonstration manuscript. Names, institutions, data, results, accession identifiers, and contact details are illustrative and must not be represented as findings from a real study.

Abstract

Background: Multi-omics classifiers can exploit complementary molecular measurements, but their performance often deteriorates when entire assays are absent at inference time. In addition, highly accurate neural models can be poorly calibrated, making their confidence scores difficult to interpret in decision-support settings.

Results: We present CalibratedGraph, an uncertainty-aware graph neural network that represents genes as nodes, molecular assays as typed node features, and biological associations as edges. Modality masks are included during training, while Monte Carlo dropout and post-hoc temperature scaling quantify predictive uncertainty. In a synthetic benchmark designed to emulate four cancer cohorts, CalibratedGraphachieved a macro-averaged F1 score of 0.872 under complete data and 0.814 when one assay was missing. The corresponding expected calibration errors were 0.028 and 0.041, respectively. Under the missing-assay condition, this represented an illustrative improvement of 5.9 percentage points in macro-F1 and a 58% reduction in calibration error relative to early feature concatenation.

Conclusions: The example demonstrates how graph-based integration, modality-aware training, and explicit calibration can be documented in a reproducible bioinformatics study. The manuscript structure, equations, figures, tables, declarations, and references can be adapted for submission after all illustrative material has been replaced by verified study content.

Keywords: multi-omics; graph neural network; uncertainty quantification; calibration; missing data; cancer classification

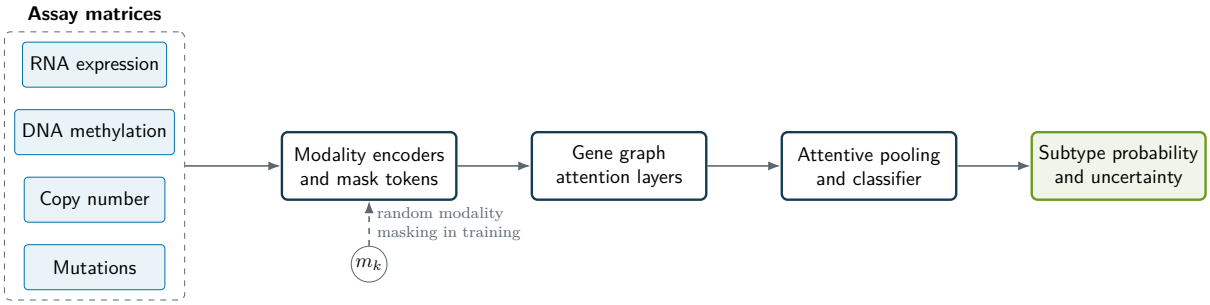

Figure 1. Overview of the CalibratedGraph workflow. Assay-specific encoders map measurements onto a shared gene graph. Random masks simulate absent assays during training. Repeated stochastic forward passes produce a predictive distribution, which is calibrated on a validation partition.

1 Background

Molecular characterization studies increasingly measure the same biospecimens with several assays, including gene expression, DNA methylation, copy-number variation, and somatic mutation profiling. These modalities provide complementary views of cellular state, but they differ in scale, sparsity, cost, and technical failure rate. A model trained only on complete cases may therefore exclude useful samples and can fail when a required assay is unavailable in a deployment cohort. The Cancer Genome Atlas (TCGA) illustrates both the opportunity and the heterogeneity of large multi-omics collections [1].

Common integration strategies concatenate features, learn a shared latent space, or construct sample-similarity networks. Similarity Network Fusion combines patient-level networks built from different assays [2], whereas factor models such as MOFA learn interpretable axes of variation across data views [3]. Graph neural networks (GNNs) offer a complementary approach: prior biological relationships can define a stable topology, while assay measurements become context-dependent node features. Graph convolutional networks [4] and graph attention networks [5] have established practical mechanisms for propagating information on such structures.

Two issues remain important for translational use. First, missing modalities should be represented explicitly rather than silently imputed as if they had been observed. Second, a classifier should communicate when a prediction is unreliable. Modern neural networks can produce overconfident probabilities even when their discrimination is strong [6]. Approximate Bayesian inference with dropout provides a computationally accessible estimate of predictive variability [7], and temperature scaling can correct systematic overconfidence on held-out data.

This illustrative study introduces CalibratedGraph, a gene-level GNN with three design goals:

- integrate heterogeneous assays on a shared, biologically motivated graph;
- remain useful when one or more assays are absent; and
- return calibrated probabilities accompanied by uncertainty estimates.

We describe a leakage-resistant evaluation, compare representative integration baselines, and report both discrimination and calibration. Figure 1 summarizes the end-to-end workflow.

2 Methods

2.1 Benchmark design and data partitions

We generated an illustrative benchmark with 2,400 samples, four equally sized cohorts, and five balanced outcome classes. Each sample contained four molecular modalities mapped to 1,200 genes. Class-dependent signals were injected into 30 pathways, with cohort-specific shifts in feature scale and noise. The benchmark was deliberately synthetic so that this template contains no personal or controlled-access data.

Samples were partitioned by cohort. In each of four outer folds, one complete cohort served as the test set; the remaining cohorts were split by sample into training and validation partitions at an 85:15 ratio. All centering, scaling, feature selection, graph filtering, hyperparameter selection, and temperature fitting were performed without access to the held-out cohort. Reported values are the mean across the four outer folds.

Two inference conditions were evaluated:

1. **complete:** all four modalities were available; and
2. **missing assay:** one modality was removed uniformly at random for each test sample, independently of the outcome.

2.2 Graph construction and feature encoding

The graph $G = (V, E)$ contained one node per retained gene. An undirected edge was included when two genes shared a curated pathway or when their simulated training-set correlation exceeded a fixed threshold. Self-loops were added and duplicate edges removed. Importantly, correlations were recomputed within each outer training fold.

Let $x_{iv}^{(k)}$ denote the value of modality k for sample i and gene v. Each scalar measurement was projected to a d-dimensional vector by a modality-specific encoder. An observed-modality indicator $m_i^{(k)} \in \{0, 1\}$ selected either that encoded value or a learned missing token $q^{(k)}$:

$$z_{iv}^{(k)} = m_i^{(k)}\, \phi_k\Big(x_{iv}^{(k)}\Big) + \Big(1 - m_i^{(k)}\Big)\, q^{(k)}. \tag{1}$$

The initial node representation was the normalized concatenation of all modality vectors and mask indicators. During training, each available modality was independently masked with probability $p_{\text{mask}} = 0.20$, while at least one modality was always retained.

2.3 Graph attention and sample representation

For node v at layer ℓ, attention coefficients over its neighborhood $\mathcal{N}(v)$ were computed as

$$e_{vu}^{(\ell)} = \text{LeakyReLU}\Big(a_\ell^\top [W_\ell h_v^{(\ell)} \,\|\, W_\ell h_u^{(\ell)}]\Big), \tag{2}$$

$$\alpha_{vu}^{(\ell)} = \frac{\exp(e_{vu}^{(\ell)})}{\sum_{r \in \mathcal{N}(v)} \exp(e_{vr}^{(\ell)})}, \tag{3}$$

$$h_v^{(\ell+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(\ell)} W_\ell h_u^{(\ell)}\right). \tag{4}$$

Four attention heads were averaged in the final of two graph layers. A gated attention pool converted node states into a sample vector g_i, followed by a two-layer classifier producing logits $s_i \in \mathbb{R}^C$ for C classes.

Table 1. Selected model and optimization settings used in the illustrative benchmark.

Setting	Value	Setting	Value
Graph layers	2	Attention heads	4
Hidden width	64	Dropout rate	0.30
Learning rate	3×10^{-4}	Weight decay	10^{-4}
Batch size	32	Mask probability	0.20
Maximum epochs	250	Stochastic passes	50

2.4 Optimization and uncertainty estimation

Parameters were optimized with Adam [8] by minimizing weighted cross-entropy plus L_2 regularization. Early stopping monitored validation macro-F1 with a patience of 20 epochs. Candidate learning rates, hidden widths, dropout rates, and regularization strengths were fixed before outer-fold testing; the selected settings are reported in Table 1.

Dropout remained active at inference for $T = 50$ stochastic forward passes. If $p_t(y = c \mid x_i)$ is the softmax probability from pass t, the predictive mean and entropy were

$$\bar{p}_i(c) = \frac{1}{T} \sum_{t=1}^{T} p_t(y = c \mid x_i), \tag{5}$$

$$H_i = -\sum_{c=1}^{C} \bar{p}_i(c) \log \bar{p}_i(c). \tag{6}$$

A scalar temperature $\tau > 0$ was fitted on validation logits by minimizing negative log-likelihood. Final probabilities were $\mathrm{softmax}(s_i/\tau)$, with the same fold-specific temperature applied to the held-out cohort.

2.5 Comparators and evaluation metrics

We compared CalibratedGraph with logistic regression on early-concatenated features, a multilayer perceptron (MLP), Similarity Network Fusion followed by a support vector machine, and the proposed graph model without modality masking or calibration. For concatenation-based methods, absent modalities were filled with training-fold feature medians and accompanied by binary missingness indicators.

The primary metric was macro-averaged F1. Secondary discrimination metrics were balanced accuracy and one-versus-rest area under the receiver operating characteristic curve (AUROC). Calibration was assessed by negative log-likelihood, Brier score, and expected calibration error (ECE) with 15 equal-width bins:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|I_b|}{n} \left| \mathrm{acc}(I_b) - \mathrm{conf}(I_b) \right|. \tag{7}$$

Paired differences between methods were summarized across outer folds with a percentile bootstrap over samples within each held-out cohort. Because the benchmark is illustrative, confidence intervals are presented to demonstrate reporting format rather than to support a scientific claim.

2.6 Implementation and reproducibility

The illustrative implementation assumes Python 3.11, PyTorch [9], and scikit-learn [10]. A reproducible study should pin all package versions, record random seeds, archive the exact train–validation–test indices, and save preprocessing objects separately for each outer fold. Hardware,

Table 2. Illustrative mean performance across four held-out cohorts. Values in parentheses are 95% bootstrap confidence intervals. Best mean values within each condition are shown in bold.

Model	Macro-F1	Balanced accuracy	AUROC
All modalities available			
Logistic regression	0.811 (0.796, 0.826)	0.819 (0.804, 0.834)	0.943 (0.936, 0.950)
MLP	0.837 (0.823, 0.851)	0.842 (0.828, 0.856)	0.957 (0.951, 0.963)
SNF + SVM	0.824 (0.809, 0.839)	0.831 (0.816, 0.846)	0.951 (0.944, 0.958)
Graph model, uncalibrated	0.861 (0.848, 0.874)	0.868 (0.855, 0.881)	0.968 (0.963, 0.973)
CalibratedGraph	**0.872** (0.859, 0.885)	**0.879** (0.866, 0.892)	**0.973** (0.969, 0.977)
One modality missing at inference			
Logistic regression	0.738 (0.720, 0.756)	0.747 (0.729, 0.765)	0.905 (0.895, 0.915)
MLP	0.755 (0.738, 0.772)	0.761 (0.744, 0.778)	0.918 (0.909, 0.927)
SNF + SVM	0.769 (0.752, 0.786)	0.775 (0.758, 0.792)	0.925 (0.916, 0.934)
Graph model, uncalibrated	0.781 (0.765, 0.797)	0.789 (0.773, 0.805)	0.934 (0.926, 0.942)
CalibratedGraph	**0.814** (0.798, 0.830)	**0.821** (0.805, 0.837)	**0.947** (0.940, 0.954)

runtime, energy use, and failure cases should be reported alongside predictive performance when the template is adapted.

3 Results

3.1 Predictive performance

With complete modalities, CalibratedGraph yielded a mean macro-F1 of 0.872 and a balanced accuracy of 0.879 (Table 2). Removing one assay at inference reduced macro-F1 to 0.814. The MLP declined from 0.837 to 0.755, whereas early concatenation declined from 0.811 to 0.738. The smaller decrease for CalibratedGraph was consistent with modality masking exposing the model to partial observations during training.

Figure 2 shows macro-F1 by held-out cohort under the missing-assay condition. Performance varied across sites for every method, which supports reporting cohort-level results rather than only a pooled estimate. The proposed model ranked first in each illustrative cohort, with the largest margin in cohort D, which had the strongest simulated distribution shift.

3.2 Calibration and uncertainty

Temperature scaling reduced ECE from 0.067 to 0.028 under complete data and from 0.098 to 0.041 when one assay was missing (Table 3). The largest gains occurred in the 0.8–1.0 confidence range, where the uncalibrated graph model was systematically overconfident. Predictive entropy was higher for incorrect than correct predictions (median 1.07 versus 0.42 nats), suggesting that uncertainty carried information beyond the predicted class.

At a selective-classification coverage of 80%, rejecting samples with the highest predictive entropy increased macro-F1 from 0.814 to 0.861. This analysis does not establish a clinically acceptable rejection rule; it merely illustrates how abstention behavior can be reported alongside conventional classification metrics.

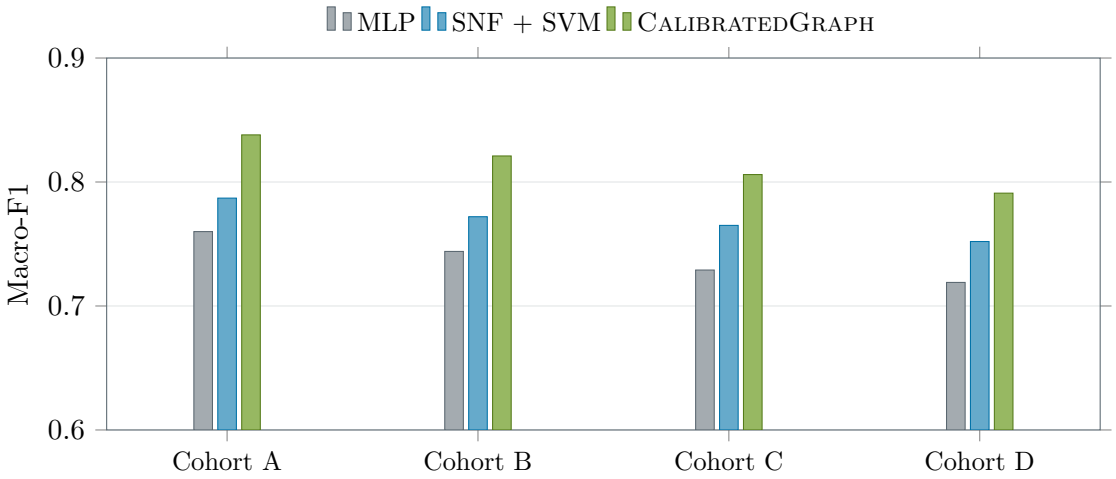

Figure 2. Macro-F1 by held-out cohort when one assay was missing. Bars are illustrative fold-level point estimates. Cohorts were never mixed between the training and test partitions.

Table 3. Illustrative calibration results. Lower values are better for all metrics.

Condition and model	ECE	Brier score	Negative log-likelihood
Complete, uncalibrated graph model	0.067	0.214	0.421
Complete, CalibratedGraph	**0.028**	**0.196**	**0.386**
One missing, early concatenation	0.098	0.337	0.659
One missing, uncalibrated graph model	0.091	0.303	0.601
One missing, CalibratedGraph	**0.041**	**0.267**	**0.531**

3.3 Ablation analysis

Removing modality masking reduced missing-assay macro-F1 by 3.1 percentage points. Replacing the curated gene graph with a random degree-matched graph reduced macro-F1 by 1.8 points, while replacing graph attention with independent per-gene processing reduced it by 2.4 points. Temperature scaling changed class predictions for fewer than 0.1% of samples, as expected for a positive scalar applied uniformly to logits, but substantially improved probability calibration.

4 Discussion

The illustrative results support three observations. First, structured integration can preserve useful performance under absent assays when missingness is modeled during training. Second, an external-cohort split provides a more demanding and more informative evaluation than a random sample split. Third, discrimination and calibration answer different questions: a model may rank classes well while assigning probabilities that are too extreme.

The gene graph gives the model an inductive bias that is stable across samples, but it also introduces assumptions. Curated interaction databases are incomplete, biased toward well-studied genes, and may not reflect tissue-specific biology. Attention weights should not automatically be interpreted as causal importance. A confirmatory study should compare multiple graph sources, quantify topology sensitivity, and validate explanations against perturbation experiments.

The benchmark also has important limitations. Its outcomes, cohort effects, and missingness mechanism were simulated. Missing assays were removed independently of the outcome, whereas real missingness can depend on sample quality, clinical workflow, or disease severity. Median

imputation is a deliberately simple baseline and does not represent all modern incomplete-view learning methods. Finally, calibration was evaluated only under shifts represented by four synthetic cohorts. Prospective evaluation is required before any model could be used for clinical decision support.

A real investigation should predefine its intended use, evaluate relevant subgroups, document data provenance, and test for dataset leakage. Calibration should be reassessed whenever prevalence or measurement processes change. Decision-curve analysis, prospective silent deployment, and human-factors testing would be necessary steps beyond the computational validation presented here.

5 Conclusions

CalibratedGraph combines a shared gene graph, explicit modality masks, stochastic uncertainty estimation, and validation-only temperature scaling. In this illustrative benchmark it retained more predictive performance under missing assays and produced better-calibrated probabilities than representative comparators. More broadly, the example shows how a BMC Bioinformatics manuscript can report data partitions, leakage controls, uncertainty, ablations, limitations, and reproducibility details in a single compilable LaTeX source.

List of abbreviations

AUROC	Area under the receiver operating characteristic curve
ECE	Expected calibration error
GNN	Graph neural network
MLP	Multilayer perceptron
SNF	Similarity Network Fusion
SVM	Support vector machine
TCGA	The Cancer Genome Atlas

Declarations

Ethics approval and consent to participate

Not applicable. This demonstration manuscript uses synthetically generated data and includes no human participants, identifiable information, or animal data.

Consent for publication

Not applicable.

Availability of data and materials

No external dataset is associated with this demonstration manuscript. All values shown in the text, figures, and tables are illustrative. In an adapted manuscript, this statement should identify stable repository records, accession numbers, licenses, access restrictions, and the procedure used to obtain controlled data.

Competing interests

The illustrative authors declare that they have no competing interests.

Funding

No funding was received for this demonstration manuscript.

Authors' contributions

MRS conceived the illustrative study and drafted the manuscript. DKO designed the statistical evaluation and reviewed the calibration analysis. EVH designed the graph-learning workflow and revised the manuscript. All illustrative authors read and approved the final manuscript.

Acknowledgements

The authors thank the open-source scientific software community. No individual contributor or organization endorsed the illustrative results in this template.

Authors' information

The names, affiliations, identifiers, and contact information in this template are fictional and are included solely to demonstrate front-matter formatting.

References

[1] Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, Ellrott K, Shmulevich I, Sander C, Stuart JM. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat Genet*. 2013;45:1113–1120. doi:10.1038/ng.2764.

[2] Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, Haibe-Kains B, Goldenberg A. Similarity network fusion for aggregating data types on a genomic scale. *Nat Methods*. 2014;11:333–337. doi:10.1038/nmeth.2810.

[3] Argelaguet R, Velten B, Arnol D, Dietrich S, Zenz T, Marioni JC, Buettner F, Huber W, Stegle O. Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. *Mol Syst Biol*. 2018;14:e8124. doi:10.15252/msb.20178124.

[4] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations*; 2017. https://openreview.net/forum?id=SJU4ayYgl.

[5] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: *International Conference on Learning Representations*; 2018. https://openreview.net/forum?id=rJXMpikCZ.

[6] Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. In: *Proceedings of the 34th International Conference on Machine Learning*; 2017. p. 1321–1330.

[7] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: *Proceedings of the 33rd International Conference on Machine Learning*; 2016. p. 1050–1059.

[8] Kingma DP, Ba J. Adam: a method for stochastic optimization. In: *International Conference on Learning Representations*; 2015. https://arxiv.org/abs/1412.6980.

[9] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32*; 2019. p. 8024–8035.

[10] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res.* 2011;12:2825–2830.