[Supplementary] AMIGO: Sparse Multi-Modal Graph Transformer with Shared-Context Processing for Representation Learning of Giga-pixel Images

Ramin Nakhli
University of British Columbia
ramin.nakhli@ubc.ca

Puria Azadi Moghadam
University of British Columbia
puria.azadi@ubc.ca

Haoyang Mi
Johns Hopkins University
hmi1@jhmi.edu

Hossein Farahani
University of British Columbia
h.farahani@ubc.ca

Alexander Baras
Johns Hopkins University
baras@jhmi.edu

Blake Gilks
University of British Columbia
blake.gilks@vch.ca

Ali Bashashati
University of British Columbia
ali.bashashati@ubc.ca

1. Evaluation Details

We used concordance index (c-index) [5] and the p-value of the LongRank test [1] to measure the performance of our model and that of the baselines.

C-index is a metric to measure the quality of ranking between the predicted and the observed survival times. More specifically, it is defined as Eq. (1)

$$c = \frac{1}{\mid \varepsilon \mid} \sum_{T_i uncensored} \sum_{T_j > T_i} \mathbf{1}_{f(x_i) < f(x_j)}, \qquad (1)$$

where $\mathbf{1}_{a<b} = 1$ if $a < b$ and 0 otherwise, $f(x_i)$ is the predicted survival time for x_i, and ε is the ordered pairs of the data points [5].

The statistical test known as the LongRank test evaluates the validity of the null hypothesis, which states that there is no difference between the survival curves of two populations at any given period. Using the median survival time predicted by the model, we split the patients into two groups of low-risk (patients with a predicted survival time greater than the median) and high-risk (patients with a predicted survival time less than the median). The p-value of the LogRank test on the survival curves of these two cohorts demonstrates the separability of the curves (p-value < 0.05).

In all the experiments, at inference time, we censor the patients with a survival time greater than 10 and 7 years for the InUIT and MIBC datasets, respectively.

2. Implementation Details

2.1. AMIGO

All the experiments were performed on a single GeForce RTX 3090 using Pytorch and DGL packages. The output feature size of the GraphSAGE layers and the MLP layers in each branch were set to 128 and 32, respectively. Adam optimizer with a learning rate of 0.002, a cosine scheduler, a weight decay of 0.0001, and a batch size of 128 were used for the training of the models. The Transformer module of the cross-modal aggregator included 4 MHSAs, and the BCP was set to 0.1. We also used a sparsity ratio of 0.8, which was only applied at training time.

2.2. Baselines

For the implementation of the baselines, we used the official repository of Patch-GCN[1], which also included the implementation for the DeepSet, Attention MIL, and DGC methods. The hyperparameters were set to the suggested values in the paper [3], and we used the NLL loss [6] as suggested. Similarly, for the Pathomic Fusion model , we also used the official implementation repositories with the suggested parameters.

For the HIPT model, the pre-trained weights and model implementation were both taken from the official repository. A loss function similar to the one we used for our model (Cox loss) was used to train an MLP on top of the representation produced by the pre-trained model. Additionally, we adopted the Adam optimizer with a learning rate of 0.001.

It is of note to mention that both the baselines and

[1] https://github.com/mahmoodlab/Patch-GCN

Ablated Feature	InUIT		MIBC	
	C-Index (↑)	P-value (↓)	C-Index (↑)	P-value (↓)
No instance Norm	0.53 ± 0.001	0.15	0.54 ± 0.005	0.03
No weight sharing	0.56 ± 0.001	0.06	0.54 ± 0.016	0.001
Full weight sharing	0.53 ± 0.001	0.46	0.51 ± 0.005	0.78
No BCP	0.54 ± 0.001	0.07	0.58 ± 0.013	0.38
Transformer attention	0.53 ± 0.002	0.05	0.55 ± 0.011	0.90
Inference-time sparsity	0.56 ± 0.001	0.04	0.55 ± 0.004	0.08
Non-shared attention	0.54 ± 0.001	0.13	0.58 ± 0.003	0.06
AMIGO (Ours)	$\mathbf{0.57 \pm 0.002}$	**0.01**	$\mathbf{0.61 \pm 0.004}$	$< \mathbf{0.001}$

Table S3. Full ablation studies of our model.

AMIGO utilize the same data sources (all images regardless of stain type). Equivalent to AMIGO's aggregator, all baseline models include attention pooling in the last layer to pick the most relevant data.

3. Heatmap Visualization

We also visualized the heatmaps of our model on the cellular graphs of a MIBC patient in Fig. S1. Interestingly, we realized that the model learns to pay more attention to the P16 stain, which is in line with previous studies in bladder cancer as they showed the importance of Ki67 in the outcomes of MIBC cases. [4].

4. Masking Visualization

The visualization of the cellular graph after the two masking operations can be found in Fig. S2. These two images are generated at two different training iterations from the same graph. We would like to highlight the variation in the sub-structural graphs within each cell graph during training, which leads to a strong augmentation and regularization for the model.

5. Full Ablation Study

The complete results for the ablation studies can be found in Tab. S3, where both c-index and p-value are reported in each case. Although all of the results support our design decisions, we found that full weight sharing of the branches considerably impairs our model's capacity to distinguish between cohorts at low and high risk on both datasets. This finding supports our assertion that a multimodal design is crucial for capturing tissue heterogeneity.

6. Effect of Normalization Type

We also compared the performance of our model with different types of normalization layers applied after the instance attention (Tab. S4). Our results demonstrate that instance normalization has a superior performance across both datasets compared to the other types.

Ablated Feature	InUIT		MIBC	
	C-Index (↑)	P-value (↓)	C-Index (↑)	P-value (↓)
No Normalization	0.53 ± 0.001	0.15	0.54 ± 0.005	0.03
Batch Normalization	0.55 ± 0.002	0.05	$\mathbf{0.62 \pm 0.002}$	0.05
Layer Normalization	0.56 ± 0.001	0.02	0.58 ± 0.010	0.54
Instance Normalization	$\mathbf{0.57 \pm 0.002}$	**0.01**	0.61 ± 0.004	$< \mathbf{0.001}$

Table S4. Effects of different normalization types after the instance attention layer on the performance of our model.

7. Cox vs NLL Loss Function

As was elaborated in Sec. 2.2, we used the NLL loss for the DeepSet, Attention MIL, DGC, and Patch-GCN as suggested by Chen *et al*. [3]. However, to ensure a fair comparison between our model and the baselines, we also measured the performance of them using the Cox loss function (Tab. S5). Similar to [3], we find that these baselines achieve a better performance with the NLL loss function.

In order to complete the comparison, we also tested our model using the NLL loss, which demonstrated worse performance compared to Cox loss.

8. Effect of BCP on Baselines

We also explored the effect of our proposed BCP technique on the baselines. Since BCP was established using the Cox loss function, the experiments were also conducted in relation to this loss (Tab. S6). As can be observed, applying the BCP technique typically improves the performance of the models, demonstrating the applicability of this technique.

9. Flops Comparison

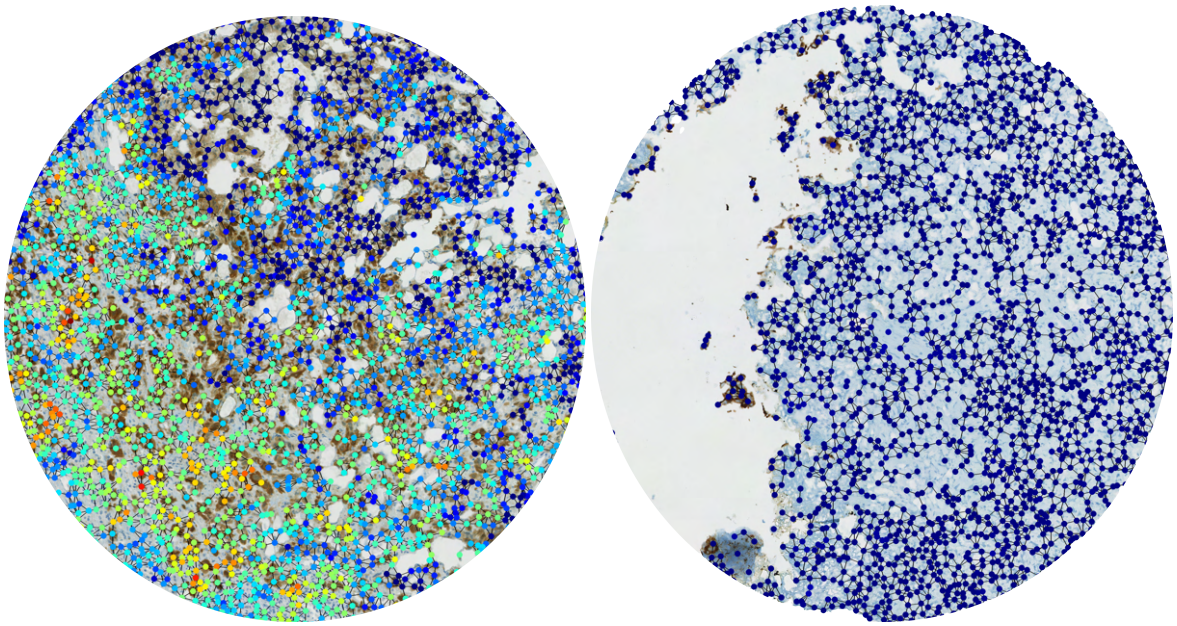
The comparison of the number of the parameters and floating-point operations (FLOPs) for the baselines alongwith that of our model can be found in Fig. S7. Although our model is a multi-modal cell-centric approach (dealing with large graphs containing hundreds of nodes), it has less parameters and FLOPs compared to the baselines.

The parameter efficiency of our model can be linked to the shared-context processing nature of it, where the major parameter bottlenecks are shared across multiple modalities. On the other hand, the FLOPs efficiency is mainly attributed to the sparse processing of our model.

10. Self-Supervised Pretraining

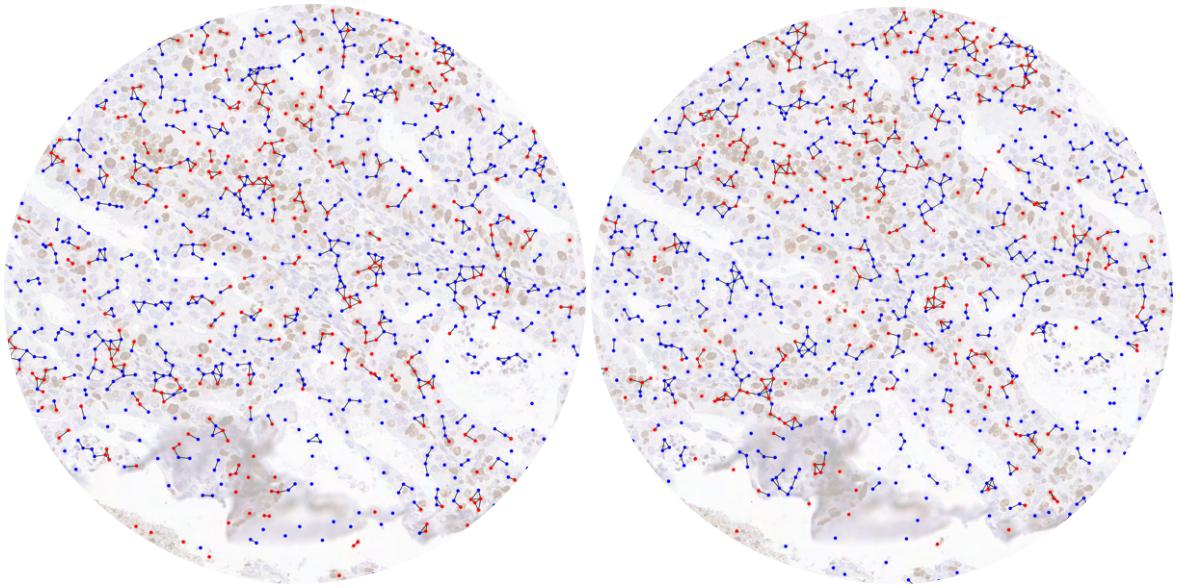
We also explored the impact of self-supervised graph representation learning methods such as BGRL [2] on the performance of our model.

Fig. S8 depicts an overview of the BGRL framework. The "online" and "target" (slow-moving average of the online model) models function together in this architecture to produce representations for the two augmentations of the



(a) P16 (b) CK20

Figure S1. Visualization of heatmap graphs for a patient with P16 and CK20 stains.



(a) Masking 1 (b) Masking 2

Figure S2. Visualization of cellular graphs after masking.

same input graph. The adopted augmentation operations include random dropping of nodes and edges of the graph, and the loss function is a cosine distance loss that brings the cosine similarity of the graph representations close to 1.

We tested this approach in two scenarios: 1) single-modal pre-training; 2) multi-modal pre-training. In the former, we only pre-train the encoders of each multi-modal branch using the BRGL framework. However, in the latter, we initialize the encoder weights with the pre-trained ones from the single-modal version and train the rest of our model (MLPs, instance attention, and the transformer model) using BGRL. Finally, in both scenarios, we fine-tuned the pre-trained model using the survival information. The results of our experiments demonstrate that the BGRL

Method	Feature Extractor	Loss	InUIT		MIBC	
			C-Index (↑)	P-value (↓)	C-Index (↑)	P-value (↓)
DeepSet	ResNet34	NLL	0.50 ± 0.0	0.43	0.50 ± 0.001	–
	ResNet34	Cox	0.50 ± 0.0	–	0.50 ± 0.0	–
	ResNet50	NLL	0.53 ± 0.007	0.40	0.45 ± 0.004	0.28
	ResNet50	Cox	0.50 ± 0.0	–	0.51 ± 0.001	–
Attention MIL	ResNet34	NLL	0.51 ± 0.004	0.62	0.59 ± 0.007	0.04
	ResNet34	Cox	0.50 ± 0.002	0.22	0.50 ± 0.002	0.76
	ResNet50	NLL	0.55 ± 0.004	0.65	0.55 ± 0.004	0.57
	ResNet50	Cox	0.51 ± 0.003	0.57	0.47 ± 0.001	0.08
DGC	ResNet34	NLL	0.53 ± 0.007	0.46	0.58 ± 0.007	$< \mathbf{0.001}$
	ResNet34	Cox	0.52 ± 0.003	0.69	0.46 ± 0.012	0.67
	ResNet50	NLL	0.55 ± 0.005	0.31	0.54 ± 0.007	0.64
	ResNet50	Cox	0.51 ± 0.001	0.75	0.50 ± 0.010	0.36
Patch-GCN	ResNet34	NLL	0.53 ± 0.008	0.45	0.50 ± 0.004	0.005
	ResNet34	Cox	0.53 ± 0.002	0.41	0.47 ± 0.005	0.58
	ResNet50	NLL	0.50 ± 0.004	0.25	0.46 ± 0.009	0.33
	ResNet50	Cox	0.52 ± 0.002	0.80	0.52 ± 0.006	0.14
HIPT	Hierarchical ViT	NLL	0.53 ± 0.001	0.87	0.46 ± 0.003	0.54
	Hierarchical ViT	Cox	0.50 ± 0.002	0.18	0.53 ± 0.010	0.10
AMIGO (Ours)	ResNet34	NLL	0.57 ± 0.003	0.02	0.57 ± 0.010	$< \mathbf{0.001}$
AMIGO (Ours)	ResNet34	Cox	$\mathbf{0.57 \pm 0.002}$	**0.01**	$\mathbf{0.61 \pm 0.004}$	$< \mathbf{0.001}$

Table S5. Comparison of the Cox and NLL loss functions for the baselines and our model.

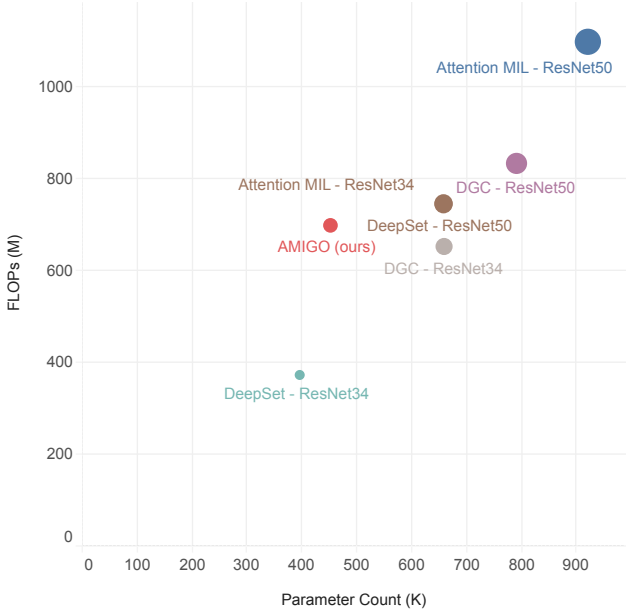


Figure S7. Parameter vs. Flops comparison of our model with the baselines. The size of the points shows is relative to the multiplication of its parameter count and FLOPs.

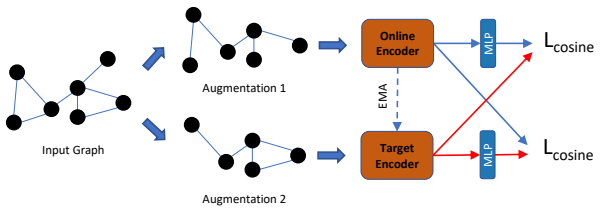


Figure S8. Self-supervised Model

self-supervised pre-training does not help achieve a consistent improvement over both datasets compared to the supervised setting (Tab. S9). We believe this shows that the self-supervised learning for histopathology graphs requires special considerations that can be a part of our future works.

References

[1] J Martin Bland and Douglas G Altman. The logrank test. *Bmj*, 328(7447):1073, 2004. 1

[2] Feihu Che, Guohua Yang, Dawei Zhang, Jianhua Tao, and Tong Liu. Self-supervised graph representation learning via bootstrapping. *Neurocomputing*, 456:88–96, 2021. 2

[3] Richard J Chen, Ming Y Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y Chen, Drew FK Williamson, and Faisal Mah-

Method	Feature Extractor	BCP	InUIT		MIBC	
			C-Index (↑)	P-value (↓)	C-Index (↑)	P-value (↓)
DeepSet	ResNet34	w/ BCP	0.50 ± 0.0	–	0.50 ± 0.001	–
	ResNet34	w/o BCP	0.50 ± 0.0	–	0.50 ± 0.0	–
	ResNet50	w/ BCP	0.53 ± 0.007	0.40	0.51 ± 0.001	–
	ResNet50	w/o BCP	0.50 ± 0.0	–	0.51 ± 0.001	–
Attention MIL	ResNet34	w/ BCP	0.55 ± 0.002	0.84	0.52 ± 0.016	0.04
	ResNet34	w/o BCP	0.50 ± 0.002	0.22	0.50 ± 0.002	0.76
	ResNet50	w/ BCP	0.52 ± 0.001	0.44	0.53 ± 0.004	0.85
	ResNet50	w/o BCP	0.51 ± 0.003	0.57	0.47 ± 0.001	0.08
DGC	ResNet34	w/ BCP	0.51 ± 0.002	0.47	0.51 ± 0.009	0.61
	ResNet34	w/o BCP	0.52 ± 0.003	0.69	0.46 ± 0.012	0.67
	ResNet50	w/ BCP	0.53 ± 0.003	0.92	0.50 ± 0.007	0.12
	ResNet50	w/o BCP	0.51 ± 0.001	0.75	0.50 ± 0.010	0.36
Patch-GCN	ResNet34	w/ BCP	0.53 ± 0.002	0.44	0.55 ± 0.009	0.07
	ResNet34	w/o BCP	0.53 ± 0.002	0.41	0.47 ± 0.005	0.58
	ResNet50	w/ BCP	0.52 ± 0.003	0.77	0.58 ± 0.009	0.16
	ResNet50	w/o BCP	0.52 ± 0.002	0.80	0.52 ± 0.006	0.14
HIPT	Hierarchical ViT	w/ BCP	0.49 ± 0.001	0.33	0.49 ± 0.001	0.89
	Hierarchical ViT	w/o BCP	0.50 ± 0.002	0.18	0.53 ± 0.010	0.10
AMIGO (Ours)	ResNet34	w/ BCP	$\mathbf{0.57 \pm 0.002}$	**0.01**	$\mathbf{0.61 \pm 0.004}$	$\mathbf{< 0.001}$
AMIGO (Ours)	ResNet34	w/o BCP	0.54 ± 0.001	0.07	0.58 ± 0.013	0.38

Table S6. Comparison of the effect of BCP on the baseline models.

Method	Pre-training Type	InUIT		MIBC	
		C-Index (↑)	P-value (↓)	C-Index (↑)	P-value (↓)
Self-supervised (BGRL)	Single-Modal	$\mathbf{0.56 \pm 0.002}$	**0.04**	0.61 ± 0.008	0.39
Self-supervised (BGRL)	Multi-Modal	0.55 ± 0.005	0.28	$\mathbf{0.62 \pm 0.011}$	**0.04**

Table S9. Comparison of self-supervised training.

mood. Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 339–349. Springer, 2021. 1, 2

[4] Haoyang Mi, Trinity J Bivalacqua, Max Kates, Roland Seiler, Peter C Black, Aleksander S Popel, and Alexander S Baras. Predictive models of response to neoadjuvant chemotherapy in muscle-invasive bladder cancer using nuclear morphology and tissue architecture. *Cell Reports Medicine*, 2(9):100382, 2021. 2

[5] Harald Steck, Balaji Krishnapuram, Cary Dehing-Oberije, Philippe Lambin, and Vikas C Raykar. On ranking in survival analysis: Bounds on the concordance index. *Advances in neural information processing systems*, 20, 2007. 1

[6] Shekoufeh Gorgi Zadeh and Matthias Schmid. Bias in cross-entropy-based training of deep survival networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):3126–3137, 2020. 1