Comparison of Digital Histology AI Models with Low-Dimensional Genomic and Clinical Models in Survival Modeling for Prostate Cancer

Supplementary Material

This is the supplementary document of our paper, entitled: "Comparison of Digital Histology AI Models with Low-Dimensional Genomic and Clinical Models in Survival Modeling for Prostate Cancer". It provides detailed information on the DPAI algorithms, hyperparameter settings, and additional supporting tables and figures.

S1. DPAI Methods Details

In this section, we briefly describe the technical details of the MIL models implemented in the paper, as they are applied to the encoded patch tokens of patient P's WSI, H_{W_P}, in order to generate the discrete hazard estimates, which we will denote $\ell_1, \ldots, \ell_4$. For simplicity, we assume we are training on only one WSI per patient, W_P, with K patches.

S1.1. Global Average

WSI tokens are first fully averaged together:

$$\mathbf{h}_P := \frac{1}{K}\sum_{k=1}^{K}\mathbf{h}_k \tag{1}$$

Before the predictor arm, two densely connected layers with node sizes of 128 and 64 with dropout are included to give this network more expressiveness, and because the other MIL methods typically compress the patch embeddings from their original dimension of 1,024.

S1.2. CLAM

CLAM [2] learns a separate attention-aggregated WSI representation for each of the four time bins. First, the patch tokens, $\mathbf{h}_k$ are compressed from dimension 1024 to 512 via linear mapping:

$$\mathbf{h}_k^{(1)} = W_1\mathbf{h}_k \tag{2}$$

Four parallel attention branches are trained with shared query and key matrices $V_a, U_a \in \mathbb{R}^{256\times512}$ and time-bin specific value matrix: $W_{a,i} \in \mathbb{R}^{1\times256}$

$$a_{i,k} = \frac{\exp\left\{\mathbf{W}_{a,i}\left(\tanh(\mathbf{V}_a\mathbf{h}_k^{(1)})\odot\sigma(\mathbf{U}_a\mathbf{h}_k^{(1)})\right)\right\}}{\sum_{j=1}^{K}\exp\left\{\mathbf{W}_{a,i}\left(\tanh(\mathbf{V}_a\mathbf{h}_j^{(1)})\odot\sigma(\mathbf{U}_a\mathbf{h}_j^{(1)})\right)\right\}} \tag{3}$$

$$\mathbf{h}_{\mathrm{P},i} = \sum_{k=1}^{K} a_{i,k}\,\mathbf{h}_k^{(1)} \tag{4}$$

Finally, a separate feed-forward predictor for each time bin, f_i, generates the conditional hazard logit:

$$\ell_i = f_i(\mathbf{h}_{P,i}) \tag{5}$$

S1.3. Patch-GCN

Patch-GCN [1] leverages graph convolutional network (GCN) layers to share information between neighboring patch representations before aggregating to a patient-level representation. Specifically, The (x, y)-coordinates of each patch in Euclidean space are fed into fast approximate k-NN $(k = 8)$ to build an adjacency matrix for the patient's WSI: $\mathbf{A}_P$. Then we can denote the patient's subgraph as $\mathbf{G}_P := (\mathbf{H}_P, \mathbf{A}_P)$.

Each GCN layer, $\mathcal{F}_{GCN}^{(l)}$, implements the following message passing functions to update the current representation of the patch nodes, $\mathbf{H}_P^{(l)} = \{\mathbf{h}_1^{(l)}, \mathbf{h}_2^{(l)}, \ldots, \mathbf{h}_K^{(l)}\}$, where for each patch v, information is shared from neighboring patch vertices, $u \in \mathcal{N}(v)$:

$$\mathbf{m}_v^{(l)} = \rho^{(l)}\left(\left\{\phi^{(l)}\left(\mathbf{h}_v^{(l)}, \mathbf{h}_u^{(l)}\right) \rightarrow \mathbf{m}_{vu}^{(l)} : u \in \mathcal{N}(v)\right\}\right) \tag{6}$$

$$\mathbf{h}_v^{(l+1)} = \zeta^{(l)}\left(\mathbf{h}_v^{(l)}, \mathbf{m}_v^{(l)}\right) \tag{7}$$

$\phi^{(l)}$ calculates each neighbor's message, $\rho^{(l)}$ aggregates the messages, and $\zeta^{(l)}$ updates the current node feature based on its aggregated message. Details of these functions are provided in the original Patch-GCN paper [1].

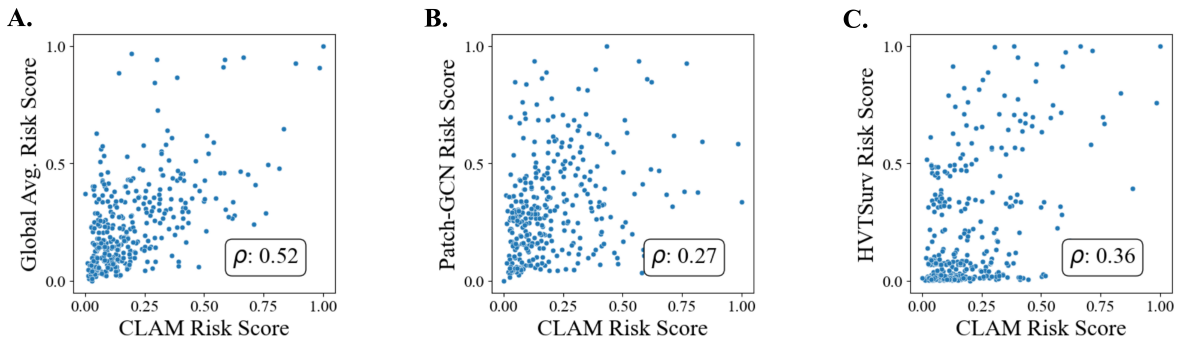

Figure S1. (A-C) Scatter plots of test set predicted risk scores on TCGA PRAD cohort for DPAI-only architectures with UNI patch encoder (A: CLAM vs Global Averaging, B: CLAM vs Patch-GCN, C: CLAM vs HVTSurv).

Four GCN layers (with residual mapping) are applied sequentially, and these four hidden layer representations are concatenated, passed through a densely connected layer to generate the final patch representations. The patch representations are aggregated with global attention-pooling to produce the patient representation $\mathbf{h}_P$.

S1.4. HVTSurv

HVTSurv [4] uses transformers at multiple hierarchical fields of view to share information between the patches. First, the patch token collection, H_{W_P}, is rearranged in chunks of length w, where the chunk includes the w nearest neighbors to the first patch in that chunk. In m different sets, a portion, $\left(\frac{m-1}{m}\right)$, of these windows are randomly masked, generating m subsets (called sub-WSIs) of unmasked, rearranged patches for the WSI.

Each sub-WSI is processed one window at a time through the "Local Window Block." This is a transformer block where the self attention (SA) in each head is biased using a learnable matrix, $\mathbf{B}$ that maps segmented Manhattan distance between patches to a relative position bias:

$$SA_{local} = softmax\left(\frac{\mathbf{QK}^\top + \mathbf{B}}{\sqrt{d}}\right) \tag{8}$$

Next, the sub-WSI patches are randomly shuffled and, one window at a time, passed through a standard self-attention transformer module. Finally, all the processed patches are concatenated and fed through a global attention pooling layer to produce the patient-level representation:

$$a_k = \frac{\exp\left\{\mathbf{U}\left(\tanh\left(\mathbf{V}\mathbf{h}_k\right)\right)\right\}}{\sum_{j=1}^{K}\exp\left\{\mathbf{U}\left(\tanh\left(\mathbf{V}\mathbf{h}_j\right)\right)\right\}}, \tag{7}$$

$$\mathbf{h}_P = \sum_{k=1}^{K} a_k \mathbf{h}_k \tag{9}$$

where $\mathbf{U} \in \mathbb{R}^{1\times d_h}$, $\mathbf{V} \in \mathbb{R}^{d_h \times d}$, and d_h is the hidden dimension. This patient-level representation can be passed to the feed-forward predictor head.

S1.5. CMTA

CMTA [5] uses encoder-decoder transformers in conjunction with cross-modal attention to fuse pathology and genomics tokens. The pathology tokens, $P = \{p_1, p_2, \ldots, p_K\}$ are first passed through a fully connected layer to reduce their dimension and then encoded using a PPEG (Pyramid Position Encoding Generator) [3] module sandwiched between two self-attention layers:

$$P^{(1)} = \text{MSA}(\text{LN}(P^{(0)})) + P^{(0)}, \tag{10}$$

$$P^{(2)} = \text{PPEG}(P^{(1)}), \tag{11}$$

$$P^{(3)} = \text{MSA}(\text{LN}(P^{(2)})) + P^{(2)} \tag{12}$$

Note $P^{(0)}$ is P but with a learnable class token appended.

The bulk RNAseq values are collected into pathway-based subsets and each is passed through a fully connected layer to generate genomics tokens that match the dimension of the patch tokens, denoted: $G = \{g_1, g_2, \ldots, g_M\}$. The genomics encoder is simply two self attention modules:

$$G^{(1)} = \text{MSA}(\text{LN}(G^{(0)})) + G^{(0)}, \tag{13}$$

$$G^{(2)} = \text{MSA}(\text{LN}(G^{(1)})) + G^{(1)} \tag{14}$$

As with $P^{(0)}$, $G^{(0)}$ is G with a learnable class token appended. The class token at the output of the last encoder layer is the intra-modal representation: p for pathology and g for genomics. Let $\mathcal{P}$ denote the instance tokens of the pathology encoder, $\{p_1^{(3)}, p_2^{(3)}, \ldots, p_K^{(3)}\}$ and $\mathcal{G}$ denote the instance tokens of the genomics encoder, $\{g_1^{(2)}, g_2^{(2)}, \ldots, g_M^{(2)}\}$. Cross-modal attention is applied to

Method	λ_{cens}	λ_{sim}	η (LR)	Epochs	Batch Size	Weight Decay	Dropout
Global Avg.	0.4		0.001	100	48	0.00001	0.1
CLAM	0.4		0.0001	20	16	0.00001	0.25
Patch-GCN	0.4		0.0005	20	16	0.00001	0.25
HVTSurv	0.4		0.0002	20	16	0.00001	0.1
CMTA	0.4	1.0	0.00005	20	16	0.01	0.1
CMTA (Gleason)	0.4	1.0	0.00005	20	16	0.01	0.1

Table S1. Hyperparameter settings for the DPAI methods.

$\mathcal{P}$ and $\mathcal{G}$ to learn interactions between the modality representations. Specifically, attention maps are computed as follows:

$$\mathcal{H}_p = \text{softmax}\left(\frac{(\mathcal{P}\mathbf{U}_p)(\mathcal{G}\mathbf{V}_p)^T}{\sqrt{d}}\right) \in \mathbb{R}^{K\times M}, \quad (15)$$

$$\mathcal{H}_g = \text{softmax}\left(\frac{(\mathcal{G}\mathbf{U}_g)(\mathcal{P}\mathbf{V}_g)^T}{\sqrt{d}}\right) \in \mathbb{R}^{M\times K}, \quad (16)$$

where $\mathbf{U}_p, \mathbf{V}_p, \mathbf{U}_g, \mathbf{V}_g \in \mathbb{R}^{d\times d}$ are learnable projections. The learnable value matrices, $\mathbf{W}_p \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_g \in \mathbb{R}^{d\times d}$, pair with the attention matrices to generate fused pathology and genomic token sets, $\mathcal{P}$ and $\mathcal{G}$, respectively:

$$\mathcal{P} = \mathcal{H}_p \times (\mathcal{G}\mathbf{W}_p) \in \mathbb{R}^{K\times d}, \quad (17)$$

$$\mathcal{G} = \mathcal{H}_g \times (\mathcal{P}\mathbf{W}_g) \in \mathbb{R}^{M\times d} \quad (18)$$

Note that some re-notating from the original paper [5] has occurred in the cross-attention definitions to ease notation for the Gleason-fusion extension.

Finally, mirrored decoders are applied to $\mathcal{P}$ and $\mathcal{G}$ to learn fused class tokens for each modality. Specifically, the pathology encoder architecture is used for $\mathcal{P}$'s decoder architecture to output genomics-fused pathology representation, $\hat{p} \in \mathbb{R}^d$. Similarly, the genomics encoder architecture is used for $\mathcal{G}$'s genomics decoder architecture to output pathology-fused genomics representation, $\hat{g} \in \mathbb{R}^d$. The final feature representation is given as:

$$\mathbf{h}_P = \left(\frac{p+\hat{p}}{2} \oplus \frac{g+\hat{g}}{2}\right) \quad (19)$$

where $\oplus$ denotes concatenation. Note that this model also applies a regularization term to enforce similarity between the intra-modal and cross-modal representations:

$$\mathcal{L}_{\text{sim}} = \frac{1}{d}\left(\|p-\hat{p}\|_1 + \|g-\hat{g}\|_1\right) \quad (20)$$

where p and g are detached from the computational graph for this loss component. The full loss is given by:

$$\mathcal{L}_{total} = \mathcal{L}_{surv} + \lambda_{sim}\mathcal{L}_{sim} \quad (21)$$

S1.6. CMTA with Gleason Fusion

We extend the CMTA model to allow for early fusion of Gleason Group with the patch and genomics tokens, rather than as a single-dimensional feature concatenation to $\mathbf{h}_P$, given that it is a highly important feature in the clinical Cox model. We treat Gleason Group as a categorical clinical variable, $c \in \mathcal{C} := \{1,2,3,4,5\}$. Each Gleason Group is mapped to a learnable embedding, $\hat{\mathbf{c}} \in \mathbb{R}^d$.

Given that c is categorical and this clinical modality is a single token, we ignore any encoder or decoder architecture for $\hat{\mathbf{c}}$. Instead, we immediately apply cross-modal attention to generate Gleason-fused pathology and Gleason-fused genomics token sets:

$$\mathcal{H}_p^{(c)} = \text{softmax}\left(\frac{(\mathcal{P}\mathbf{U}_p)(\hat{\mathbf{c}}\mathbf{V}_p)^T}{\sqrt{d}}\right) \in \mathbb{R}^{K\times 1}, \quad (22)$$

$$\mathcal{H}_g^{(c)} = \text{softmax}\left(\frac{(\mathcal{G}\mathbf{V}_g)(\hat{\mathbf{c}}\mathbf{U}_p)^T}{\sqrt{d}}\right) \in \mathbb{R}^{M\times 1}, \quad (23)$$

$$\mathcal{P}^{(c)} = \mathcal{H}_p^{(c)} \times (\hat{\mathbf{c}}\mathbf{W}_p) \in \mathbb{R}^{K\times d} \quad (24)$$

$$\mathcal{G}^{(c)} = \mathcal{H}_g^{(c)} \times (\hat{\mathbf{c}}\mathbf{W}_g) \in \mathbb{R}^{M\times d} \quad (25)$$

For model parsimony, these operations share cross-attention matrix weights ($\mathbf{U}_p$, $\mathbf{U}_g$, etc.) from the pathology and genomics fusions in Sec. S1.5. The pathology and genomics decoders from Sec. S1.5 are also reused to extract Gleason-fused pathology representation, $\hat{p}^{(c)}$, and Gleason-fused genomics representation, $\hat{g}^{(c)}$. Now, the final feature representation is given as:

$$\mathbf{h}_P = \left(\frac{p+\hat{p}+\hat{p}^{(c)}}{3} \oplus \frac{g+\hat{g}+\hat{g}^{(c)}}{3}\right) \quad (26)$$

Additionally, the representation similarity penalty is updated as follows:

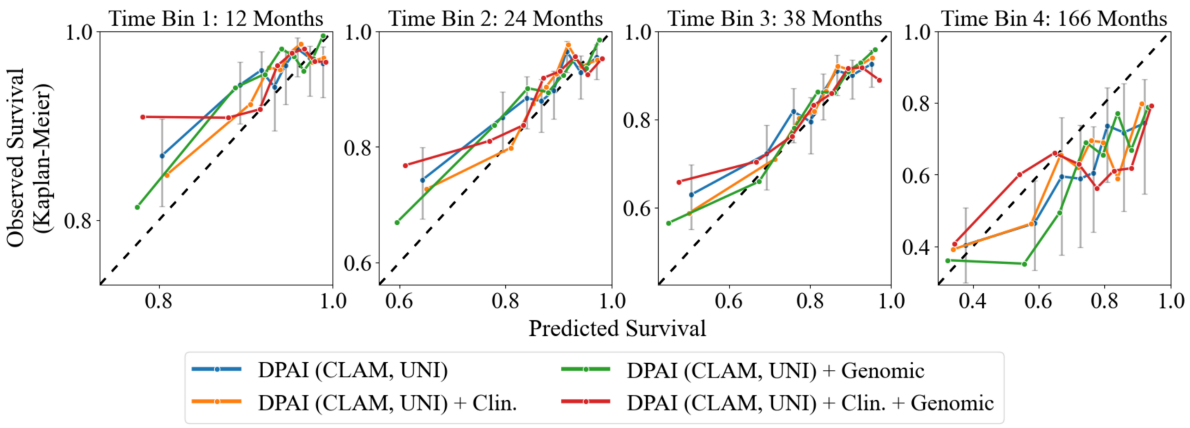

Figure S2. Calibration plots of CLAM with UNI and its multimodal counterparts. Calibration plots are computed separately for each of the four discrete time bins in which survival probabilities are predicted by the networks. The predicted survival probabilities from test sets are binned into deciles, and the Kaplan-Meier estimate from patients in each decile risk bin constitutes observed survival for that decile.

$$\mathcal{L}_{\text{sim}} = \frac{1}{d}\left(\|p-\hat{p}\|_1 + \|p-\hat{p}^{(c)}\|_1 + \|g-\hat{g}\|_1 + \|g-\hat{g}^{(c)}\|_1\right), \tag{27}$$

with p and g detached from the computational graph.

S2. Implementation Details

All models are implemented in PyTorch, with code for the models available at https://github.com/aidantmcloughlin/prad_dpai_survival. Each DPAI model uses a 2-layer predictor head 16 and 8 nodes. Adam optimizer is used. Validation set is 20% of the training set size, and the test set model corresponds to the epoch with minimal validation loss. Tab. S1 provides tuned model-specific hyperparameter settings.

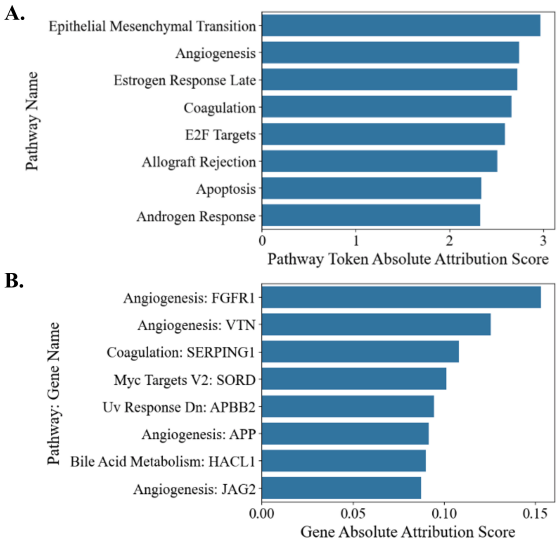

Figure S3. Absolute attribution scores of CMTA model inputs for the highest predicted risk patient in example test set (C-Index: 92%). (A) L1 norm of learned pathway tokens, top 8 scores. (B) Gene RNAseq inputs, top 8 scores. Attribution is computed using integrated gradients (IG) method.

References

[1] Richard J Chen, Ming Y Lu, Muhammad Shaban, Chengkuan Chen, Tiffany Y Chen, Drew FK Williamson, and Faisal Mahmood. Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pages 339–349. Springer, 2021. 1

[2] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021. 1

[3] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems*, 34:2136–2147, 2021. 2

[4] Zhuchen Shao, Yang Chen, Hao Bian, Jian Zhang, Guojun Liu, and Yongbing Zhang. Hvtsurv: Hierarchical vision transformer for patient-level survival prediction from whole slide image. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2209–2217, 2023. 2

[5] Fengtao Zhou and Hao Chen. Cross-modal translation and alignment for survival analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21485–21494, 2023. 2, 3