Lung Tumor Classification: differences between Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC)

Caterina Belluti,[1] Sara Gherardi,[2]
University of Modena and Reggio Emilia
Emails: {300357[1], 306432[2]}@studenti.unimore.it

***Abstract*—The objective of this paper is to present our deep learning solution to classify different types of cancer. We developed it focusing on lung tumor, our use case being between Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC), but it can be generalized for different datasets. We adopted a graph-classification approach, representing each patient as a graph where nodes correspond to genes, while edges encode functional correlations between genes derived from their coded proteins (PPI) relationships. To perform this task, we considered an architecture capable of integrating patient multi-omics data, such as mRNA gene expression, DNA methylation beta-values and copy number variations (CNV), along with clinical features given by the patient lifestyle. To obtain the best performance, we tried different architectures, such as simple Multilayer Perceptrons (MLPs), Graph Neural Networks (GNNs) and Graph Attention Networks (GATs). The model is designed to learn which genes and clinical features play a predominant role in distinguishing between these two classes of lung cancer. Our experimental results show that the integration of multi-omics and clinical data through a unified multimodal architecture successfully allows it to classify the two cancer types and retrieve which features are worth focusing on. Our code is available on GitHub. [1]**

I. Introduction

We were provided a dataset of patients affected by either LUAD or LUSC lung cancer: for each one there were biological data files containing mRNA genes expression values, Copy Number Variation (CNV) values and Methylation beta values. Another file listed the patients' clinical features, such as personal information and lifestyle habits.

We extracted the expression values of the human genes present in the analysis files, retrieving for each also their CNVs and eventual Methylation data. Starting from those, and obtaining the probability of interactions between the genes, we created graphs representing each patient's genomic overview. As late integration to the graphs, we considered the clinical features of the patients to be elaborated in a second version of the architecture. We took into consideration only those useful to our task and for which the majority of the values were present in the records.

Our pipeline is shown in Figure 1; now we will talk more in detail about each step.

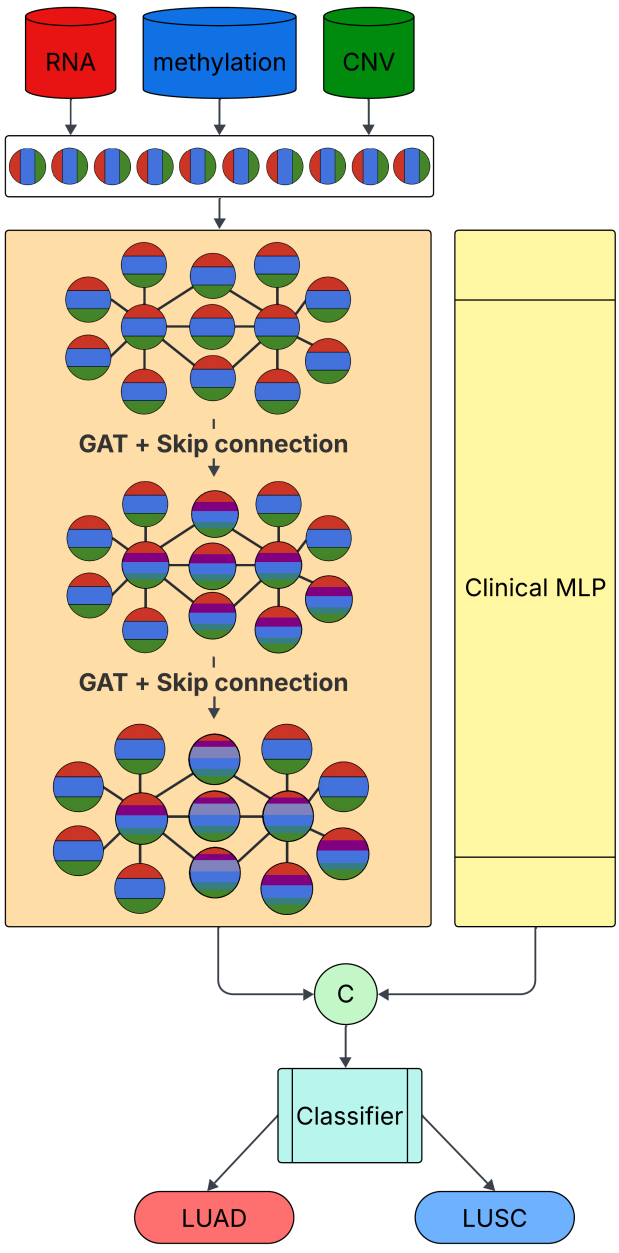

Fig. 1. Architecture

A. Related work

Previous studies have explored the use of graph neural networks for multi-omics cancer classification tasks. In particular, Zhang et al. [2] proposed a multimodal GNN framework integrating heterogeneous omics interactions for cancer molecular subtype prediction. Recent studies have also investigated graph neural networks for multi-omics integration in lung cancer analysis. In particular, Lin et al. [3] proposed MoAGNN, a hierarchical graph neural network framework integrating transcriptomic, DNA methylation, CNV, and miRNA data for LUAD subtype classification and prognosis prediction using TCGA datasets. Their results demonstrated the effectiveness of biologically informed graph-based approaches for cancer classification tasks.

II. Data

A. Dataset

The dataset was retrieved from GDC Data Portal [4], obtaining 709 patients affected by either Lung Adenocarcinoma (LUAD, 316 patients) or Lung Squamous Cell Carcinoma (LUSC, 393 patients). To start with our analysis, we kept only the TSV files containing RNA gene expression values, gene Copy Number Variations and Methylation beta-values, matching them with the patient they referred to. The patients were univocally identified through the files by their Case ID; a JSON file attached to the downloaded data was used to find the associations and map each data file to the correct patient. Later, we filtered out the patients with not exactly three associated omic files.

B. Additional sources

We downloaded the following files from the STRING database [5]:

- `9606.protein.aliases.v12.0.txt`: contains human protein isoforms associated with name aliases of their coding genes.
- `9606.protein.links.v12.0.txt`: contains pairwise protein–protein interaction links for human proteins.

With the help of the MyGene [6] python library, the first file was useful in creating a new TSV file that maps every human protein (and its codifying gene names, subjected to variations over time) to each unique gene ID (ENSG identifier). This file is mainly used through the procedure to lead-back any possible gene name to the gene unique ID, since different names were used in different data files, and some discrepancies could be found. It was also used to create a file containing a unique mapping between gene IDs and ordered integers, used later to simplify graph data management.

We also downloaded the file `humanmethylation450_15017482_v1-2.csv`, an Illumina HumanMethylation450 BeadChip manifest related to the "450K array" technology [7], which was used for methylation data preprocessing.

C. Preprocessing

Before using the different data modalities obtained from the dataset, each one goes through a preprocessing phase so that, at the end, we will have clean and meaningful features to use for our architecture.

1) RNA data: Each file contains more data columns, but we consider only `gene_id`, `gene_name`, `gene_type`, and `tpm_unstranded`.

TABLE I
SUBSET OF GENE EXPRESSION FROM RNA-SEQ TSV FILES

gene_id	gene_name	gene_type	tpm_unstranded
ENSG00000000003.15	TSPAN6	protein_coding	42.38
ENSG00000207164.1	Y_RNA	misc_RNA	0.0000
ENSG00000089060.11	SLC8B1	protein_coding	19.2524

Given a patient, we are interested in retrieving the expression value of relevant genes. We start by cleaning the current file by removing any possible metadata and keeping only the rows related to genes whose type is 'protein coding'. As expression feature, we decided to keep the Transcripts Per Million (TPM [8]) value, indicative of the scaled number of transcripts seen in a sample. After filtering, the TPM-unstranded feature is transformed using the formula

$$log(1 + x) \tag{1}$$

in order to have limited but relevant differences between values.

Then, using the file created to map genes and proteins, we find the proteins codified by the genes we have for the specific patient. The second file downloaded from STRING [5] contains the probability of interaction between any two proteins; we chose to consider only those with a probability greater than 0.7 and used it to determine the likelihood of any pairing of a patient's genes to be correlated with each other. Those are saved in a second dataframe that will be used to create the edges of the graph for each patient.

2) CNV data: The columns read from the files are `gene_id`, `gene_name`, `copy_number`, `min_copy_number` and `max_copy_number`.

TABLE II
SUBSET OF GENE EXPRESSION FROM CNV TSV FILES

gene_id	gene_name	copy_number	min_copy_number	max_copy_number
ENSG00000240361.2	OR4G11P	2	2	2
ENSG00000266580.1	MIR4254	3	3	3
ENSG00000198786.4	EGFR	4	2	6

The last two columns represent the minimum and maximum values observed in all genomic segments that overlap the gene. In the dataset, each row corresponds to a genomic segment associated with that gene. A larger difference between these values indicates a higher level of local genomic instability. To quantify this instability, we compute the following measure for each segment:

$$\Delta CNV_{ij} = CNV_{ij}^{max} - CNV_{ij}^{min} \tag{2}$$

where CNV_{ij}^{max} and CNV_{ij}^{min} are the maximum and minimum copy number values for segment j overlapping gene i.

Due to multiple overlapping segments, multiple rows may correspond to the same gene, so we aggregate the data at the gene level. For each gene, the final copy number value is computed as the median of the observed copy numbers:

$$CNV_i = \text{median}\left(\{CNV_{ij}\}_{j=1}^{n_i}\right) \tag{3}$$

where n_i is the number of segments associated with gene i.

The instability score for each gene is defined as the maximum difference observed among its segments:

$$CNV_i = \max\left(\{\Delta CNV_{ij}\}_{j=1}^{n_i}\right) \tag{4}$$

This aggregation provides a robust estimate of the gene-level copy number while preserving the strongest signal of structural variation affecting the gene.

The resulting CNV feature vector associated with each gene node is defined as $x_i^{CNV} = [CNV_i, \Delta CNV_i]$

3) Methylation data: The files contain only two data columns:

- the DNA region where methylation is applied, with respect to CpG islands
- the beta-value, which indicates how much the site is methylated

TABLE III
SUBSET OF CPG METHYLATION VALUES FROM TXT FILES

cpg_IlmnID	beta_value
cg00000029	0.117452291690285
cg00000108	0.966631960822455
cg00000807	NA

We started by keeping only data strictly related to CpG islands, filtering out the rows that did not match the format. Then we used the Illumina "450 K array" manifest [7], previously downloaded, to retrieve which genes correspond to the methylation sites and therefore might have their expression affected. The CpG islands for which no correspondence was found in the manifest are ignored. We considered only the sites relative to the promoter regions of genes, filtered based on the value of the `cpg_region` column. We found some genes to have multiple correspondence, so we proceed to group them by Id and compute the weighted average of the beta-values based on the position of the methylation region relative to the CpG island [9]. That's because the methylation effects are more relevant the closer it is to the center of the promoter site, but we decided to consider also "shores" and "shelves", with a lower impact. Therefore, we assigned a different weight to each CpG probe based on its genomic position, specified in the `cpg_position` column:

$$w_{ij} = \begin{cases} 1.0 & \text{if CpG}_{ij} \text{ is in an Island} \\ 0.6 & \text{if CpG}_{ij} \text{ is in a Shore} \\ 0.2 & \text{if CpG}_{ij} \text{ is in a Shelf} \end{cases}$$

where j denotes the CpG probe associated with gene i.

This information too is retrieved from the manifest. Each gene found will therefore have a single beta-value representing its related amount of methylation. The impact of doing the weighted average is shown in Table X, under the Evaluation section. While testing the MultiModalGNN architecture, the mean Validation Loss computed across the 5-folds resulted in 0.530 when taking into account only the Islands cpg-positions, and 0.403 considering the weighted average.

4) Clinical data: In addition to genomic data, the downloaded dataset included two clinical files: one containing general patient and tumor information, and another describing patient exposure to potentially harmful substances. Features with more than 50% missing values are discarded to reduce noise and improve reliability.

We decided to process and aggregate Tobacco-related variables into more informative features. Specifically, the smoking onset year, smoking cessation year, and smoking status were combined to derive:

- a binary feature indicating whether the patient had a history of tobacco exposure or not (True/False)
- a numerical value indicating the smoking duration computed as the difference between `exposures.tobacco_smoking_quit_year` and `exposures.tobacco_smoking_onset_year` whenever both values were available.

Patients explicitly labeled as lifelong non-smokers are encoded accordingly, while incomplete or ambiguous smoking information is treated as missing data. In datasets where smoking onset or cessation information is unavailable, these columns are not generated.

For tumor-related clinical information, we decided to retain all non-discriminative features while excluding identifiers and variables directly revealing the target classes. Categorical variables were converted into numerical representations through boolean mapping or one-hot encoding, whereas ordinal clinical stages were encoded using progressive integer values. The resulting processed clinical dataset, together with the LUAD/LUSC class label, was indexed by patient Case ID and stored for later stages of the pipeline.

TABLE IV
SUBSET OF ROWS AND COLUMNS FROM THE MERGED TSV CLINICAL FILES

project_id	pack_years_smoked	gender	sites
TCGA-LUSC	60.0	male	Peripheral Lung
TCGA-LUAD	45.0	male	Peripheral Lung
TCGA-LUSC	20.0	female	Central Lung

III. APPROACH

Our objective was to create a graph for each patient using relevant genes as nodes and their interactions as edges, accompanied by the clinical features associated. For faster execution, we decided to create our own Torch Geometric Dataset of patients, saved in three different folders for training, validation and testing of the model. Later, we combined a

GAT2ConvGNN, that takes as input the graph data, with a simple MLP, which processes the clinical data, to cooperate in finding the genes and features more relevant in the classification of the two tumor types.

A. Graph Dataset Creation

We define the class structure of the Data objects composing our Dataset. Each element represents a patient, with a part that stores the graph data and another for the clinical data, this one retrieved directly from the previously processed file. Each possible human gene is mapped to a numerical ordered index, which will identify it through the different graphs and is necessary to store the edges as couples of IDs of nodes connected. Every patient has the same number of nodes, represented by tensors: the genes not relevant for a specific patient will have all feature values set to zeroes. First, we retrieve the specific patient genes from the RNA preprocessing script, along with a Dataframe containing their probable interactions. Then, the selected genes data is integrated with the values extracted from the patient's CNV preprocessing. Lastly, we also add the methylation beta-values, retrieving them from the methylation preprocessing and adding a boolean mask to signal if it's present for a specific gene or not. These data are used to create a Tensor containing the features of the nodes. The Dataframe with their interactions is used to retrieve the Edge Index in the form of a two dimensional Tensor containing nodes pairs, and also to compute additional edges features, such as:

- the `average_combined_score` of the probabilities of interaction between the protein isoforms of the two genes connected
- the `maximum_combined_score`, since in case there is one couple of protein isoforms much more likely to interact than another between the same two genes, it will be more indicative
- `num_prot_isoforms_links`, the number of isoforms that interact with each other between the two genes; it raises the overall probability.

A Torch Geometric graph Data object is then created using these components and stored in the Dataset element. A variable containing the graph label and a Tensor with the clinical values retrieved from the features file are also created and stored.

B. Feature Scaling

Before passing the data to the model, scaling is required because the data comes in different formats and ranges between features. All the node features are scaled using a Sklearn StandardScaler fitted on the defined train Dataset. We decided to also use a MinMaxScaler to transform edge attributes plus another StandardScaler to process the clinical features with high ranges of values, for example the age of the patient. The same scalers are then used to transform the validation and test Datasets.

C. Deep Learning Model

For the model, we tried to integrate a Graph Neural Network for the analysis of the gene interactions modeled as graphs with a simple Multi-Layer Perceptron to process the clinical input-data of the patients. Our aim was to observe whether possible external factors might play an important role in the biological distinction between the two classes of tumors (LUAD - LUSC).

1) GATConvGNN: We used an architecture based on the GATv2Conv operator from the Torch Geometric library, which allows the model to grasp the relations between nodes and their neighbors using graph convolution. The branch is composed of two layers with skip-connections to preserve the signal and prevent vanishing. The first layer uses two attention heads, and edge features are integrated in the computation. To catch different aspects, the graph embedding returned is composed by concatenating a mean-pooling of the signal with its max-pooling: it was observed that it increases prediction accuracy significantly, allowing it to surpass 80% and reach 90%.

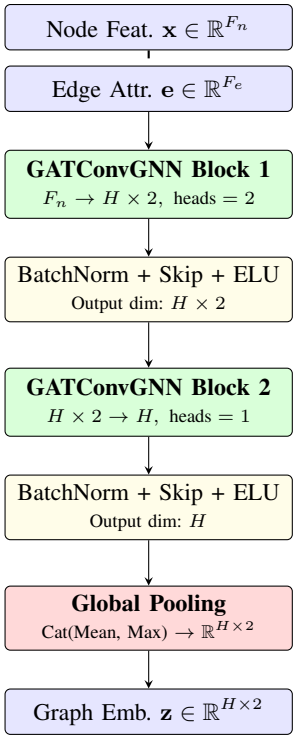

Fig. 2. Architecture of the GATConvGNN branch.

2) MLP: The clinical branch processes data through a series of non-linear transformations. Regularization is performed by Dropout layers and Layer Normalization to prevent overfitting. We adopt the Sigmoid Linear Unit (SiLU) activation instead of the Rectified Linear Unit (ReLU) due to its smoother, non-monotonic behavior, which improves gradient flow and training stability. The formula of the SiLU is:

$$\mathrm{SiLU}(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \tag{5}$$

where x is the input and $\sigma(x)$ is the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

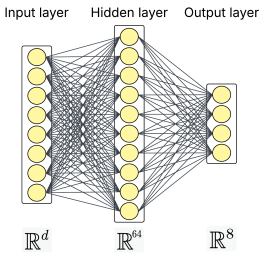

Fig. 3. Architecture of the MLP-based clinical branch. The network processes d input patient features through a sequential bottleneck structure. Each linear block consists of a fully connected layer followed by Layer Normalization and SiLU activation to stabilize training. Dropout layers are placed to prevent overfitting and improve generalization on clinical data.

3) MultiModal: The unified model takes the outputs of the GATConvGNN and MLP branches. Let $\mathbf{h}_g$ and $\mathbf{h}_c$ denote the graph and clinical embeddings, respectively. Before fusion, both embeddings are L2-normalized in order to bring them to the same scale:

$$\mathbf{h}'_g = \frac{\mathbf{h}_g}{\|\mathbf{h}_g\|_2}, \qquad \mathbf{h}'_c = \frac{\mathbf{h}_c}{\|\mathbf{h}_c\|_2} \tag{7}$$

where $\|\cdot\|_2$ denotes the L2 norm defined as

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{k=1}^{d} x_k^2}. \tag{8}$$

The normalized embeddings are then concatenated in order to combine graph-based and clinical information.

Before combining the embeddings, a gating mechanism (a sub-net with sigmoid activation) can be added to compute a [0-1] coefficient used to scale the importance of the clinical features for the specific patients, to give more or less relative weight to the graph features. The final embedding is then passed to a linear classifier which performs the class predictions.

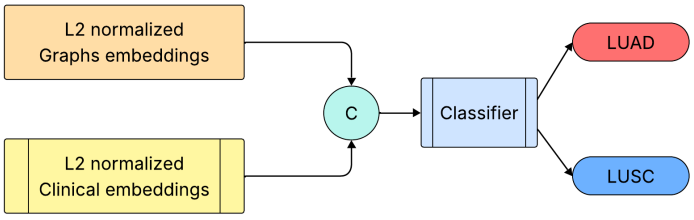

Fig. 4. MultiModal architecture

IV. Evaluation

Let's now consider the performances of our system that helped in the choice of the best model configuration for our architecture. Here are the formulas of the metrics that we used:

$$precision = \frac{TP}{TP + FP} \tag{9}$$

and

$$recall = \frac{TP}{TP + FN} \tag{10}$$

with TP = True Positives, FP = False Positives and FN = False Negatives. In our setting, LUAD corresponds to class 0 (negative class), while LUSC corresponds to class 1 (positive class).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{12}$$

We also used:

- ROC AUC (Area Under the Receiver Operating Characteristic curve) to assess the model's ability to discriminate between classes across all possible thresholds. The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR):

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

$$FPR = \frac{FP}{FP + TN} \tag{14}$$

A. Model Performance

To evaluate the performance of the MultiModalGNN architecture, we employed a 5-fold Stratified Cross-Validation approach; this ensures that each fold maintains the same percentage of samples for each class as the complete set. For each fold, we computed a set of metrics to assess both the classification accuracy and the reliability of the model. We report the final results as the mean and standard deviation across all folds to account for model stability.

TABLE V
K FOLD FOR MULTIMODAL

K	Accuracy	Precision	Recall	F1-score	ROC AUC
1	0.92	0.95	0.90	0.92	0.95
2	0.92	0.95	0.91	0.93	0.96
3	0.93	0.92	0.96	0.94	0.98
4	0.89	0.96	0.85	0.90	0.97
5	0.92	0.95	0.91	0.93	0.96

We also computed separately for each fold TP, FP, FN and TN to construct the confusion matrix:

TABLE VI
CONFUSION MATRIX K-FOLD MULTIMODAL

		Predicted	
		LUAD (0)	LUSC (1)
Real	LUAD (0)	0.93	0.07
	LUSC (1)	0.09	0.91

The global confusion matrix across the five folds reveals a well-balanced classification performance, with 93% of LUAD (Class 0) samples and 91% of LUSC (Class 1) samples correctly classified. The balanced distribution of misclassifications indicates that the architecture does not exhibit a significant bias toward either lung cancer class.

We also adopted a Monte Carlo Cross-Validation strategy with 30 independent iterations and computed the same metrics. The summary of the final results are reported as the mean values across all iterations, together with the corresponding 95% confidence intervals, to account for model stability and variability.

TABLE VII
MONTE CARLO RESULTS FOR MULTIMODALGNN

Accuracy	F1-score	ROC AUC
0.91 ± 0.03	0.92 ± 0.02	0.97 ± 0.02

Now we consider the k-folds comparison of our MultiModalGNN proposal against other approaches, evaluating the performance differences regarding the graph-classification task and the eventual gain in considering also the clinical features. In particular, we used our Dataset to test:

- A simple MLP architecture [10] taking as input a vector given by the global mean pooling of a graph's features across its nodes.
- A simple GCN network implementing the GCNConv [11] operator from the Torch Geometric library.
- A GINEConv-based GNN which implements the GINEConv [12] operator from the Torch Geometric library, an upgraded version of the GINConv operator that takes as input also edge-features.
- The MoAGNN architecture [13] a gene-centric hierarchical graph neural network integrating mRNA expression, methylation, and CNV data within a biologically informed framework.
- A basic GraphConv-based GNN which implements the GraphConv [14] operator from the Torch Geometric library.
- Our GATConv-based GNN, the MultiModalGNN graph-branch considered as a stand-alone architecture, implementing the GATv2Conv [15] operator from the Torch Geometric library.

TABLE VIII
COMPARISON K-FOLD

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
MLP	0.66 ± 0.02	0.66 ± 0.02	0.64 ± 0.03	0.63 ± 0.04	0.73 ± 0.04
GCN	0.68 ± 0.01	0.69 ± 0.02	0.67 ± 0.01	0.68 ± 0.01	0.75 ± 0.01
GINEConvGNN	0.76 ± 0.05	0.75 ± 0.05	0.86 ± 0.11	0.80 ± 0.05	0.83 ± 0.03
MoAGNN	0.88 ± 0.07	0.88 ± 0.07	0.88 ± 0.08	0.88 ± 0.08	0.93 ± 0.05
GraphConvGNN	0.89 ± 0.01	0.89 ± 0.01	0.89 ± 0.01	0.89 ± 0.01	0.95 ± 0.01
GATConvGNN	0.94 ± 0.02	0.97 ± 0.02	0.92 ± 0.02	0.94 ± 0.02	0.97 ± 0.01
MultiModal	0.92 ± 0.01	0.94 ± 0.02	0.91 ± 0.04	0.92 ± 0.01	0.96 ± 0.01

It should be noted that, for the LUAD vs LUSC classification task, clinical features have limited relevance, and therefore might add noise. When the MLP branch was evaluated independently, its accuracy rarely exceeded 0.70 and showed a strong tendency to overfitting, although this effect was partially mitigated during training. While the clinical branch contributes marginally to improving the stability of the multimodal architecture, the graph-based branch alone, based on the GATConvGNN architecture, is able to achieve accuracy scores higher than 0.90.

We performed grid-search parameters analysis to justify our architectural choices, testing different learning rates and number of hidden channels across the architectures. Here we report the loss score of each configuration, highlighting the best ones (the lower the better). We can notice that the MultiModalGNN configuration with Learning Rate = 0.001 and Hidden Channels = 64 showed the best validation loss score of all possible combinations and models.

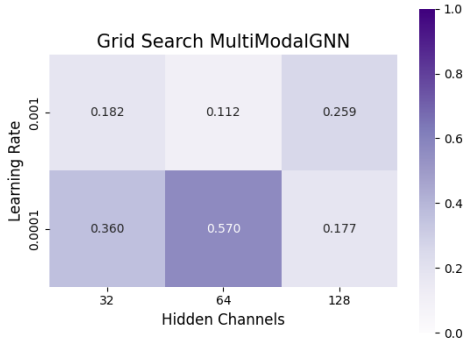

Fig. 5. HeatMap showing the best validation loss found for each combination of parameters tested applied to MultiModalGNN.

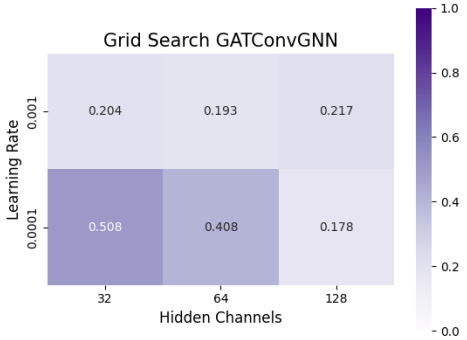

Fig. 6. HeatMap showing the best validation loss found for each combination of parameters tested applied to GATconvGNN.

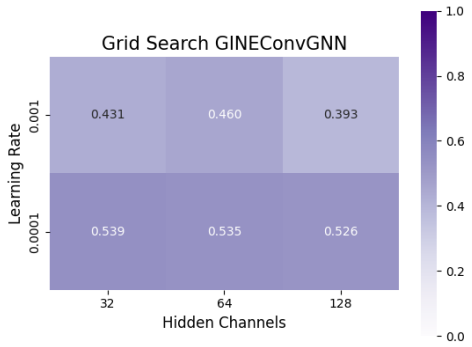

Fig. 7. HeatMap showing the best validation loss found for each combination of parameters tested applied to GINEConvGNN.

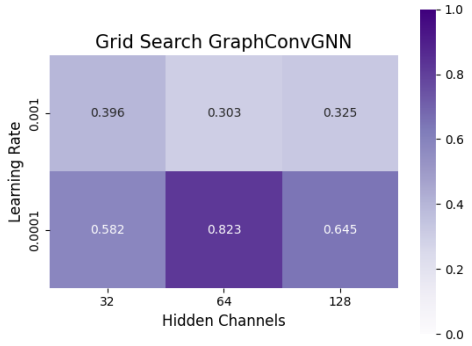

Fig. 8. HeatMap showing the best validation loss found for each combination of parameters tested applied to GraphConvGNN.

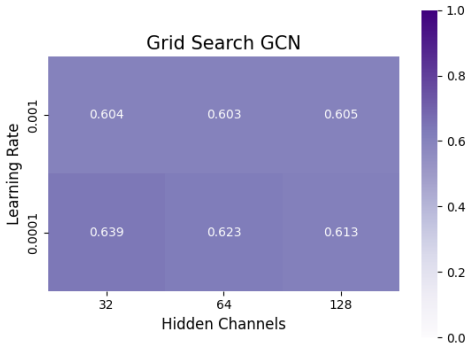

Fig. 9. HeatMap showing the best validation loss found for each combination of parameters tested applied to GCN.

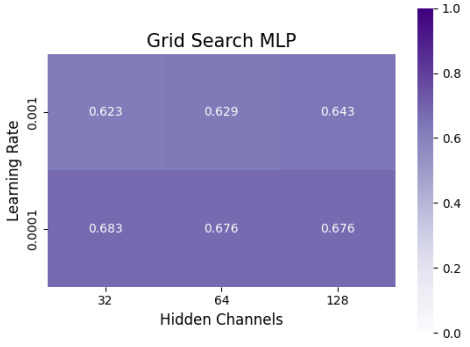

Fig. 10. HeatMap showing the best validation loss found for each combination of parameters tested applied to the simple MLP architecture.

To further compare the different architectures, we performed statistical analyses on the results obtained across the k-folds. In particular, we applied the paired T-test as a parametric test and the Mann-Whitney U-test as a non-parametric alternative to evaluate whether the observed differences between models were statistically significant. The results show that the GATConv-based architectures achieved the best performances across most evaluation metrics. Both GATConvGNN and MultiModalGNN significantly outperformed the GraphConv, GINEConv, and MLP approaches, especially in terms of accuracy, precision, F1-score and ROC AUC. In contrast, no statistically significant differences were observed between the GATConv and MultiModal architectures, indicating comparable predictive performance between the two models. Consequently, while the two models can be considered statistically comparable in terms of overall performance, the MultiModal architecture demonstrates a practical advantage also for different datasets and tumors. These results further confirm that graph-based architectures represent a suitable choice for this task, as they are able to effectively model the complex relationships between biological entities and exploit the structural information contained in the interaction networks.

We also compared the proposed architectures with MoAGNN [13], a biologically informed hierarchical graph neural network for multi-omics integration. In particular, we evaluated MoAGNN on our LUAD/LUSC classification task using the same dataset and the same biologically informed graphs adopted in our framework, in order to ensure a fair and consistent comparison between the evaluated architectures. Both GATConvGNN and MultiModalGNN achieved better overall performances than MoAGNN across most evaluation metrics, with GATConvGNN also showing statistically significant improvements for several metrics according to the Mann–Whitney U-test. These results suggest that the proposed GATConv-based architectures are more effective for the LUAD/LUSC classification task considered in this work.

Table IX summarizes the statistical comparisons between the evaluated architectures. The best-performing model is highlighted in bold, while statistically significant results ($p < 0.05$) are marked with *.

TABLE IX
STATISTICAL COMPARISON BETWEEN MODELS USING T-TEST AND U-TEST (p-VALUES)

Comparison	Metric	Paired T-test	Mann–Whitney U-test
GATConvGNN vs GraphConvGNN	Accuracy	0.024*	0.016*
	Precision	0.002*	0.012*
	Recall	0.213	0.093
	F1-score	0.014*	0.012*
	ROC AUC	0.013*	0.008*
MultiModalGNN vs GraphConvGNN	Accuracy	0.018*	0.045*
	Precision	0.009*	0.012*
	Recall	0.575	0.293
	F1-score	0.008*	0.036*
	ROC AUC	0.129	0.056
MultiModalGNN vs GATConvGNN	Accuracy	0.186	0.115
	Precision	0.091	0.094
	Recall	0.665	0.830
	F1-score	0.214	0.142
	ROC AUC	0.129	0.222
GraphConvGNN vs MoAGNN	Accuracy	0.688	0.292
	Precision	0.686	0.292
	Recall	0.635	0.292
	F1-score	0.663	0.292
	ROC AUC	0.561	0.310
MultiModalGNN vs MoAGNN	Accuracy	0.346	0.243
	Precision	0.121	0.021*
	Recall	0.554	1.000
	F1-score	0.276	0.094
	ROC AUC	0.277	0.421
GATConvGNN vs MoAGNN	Accuracy	0.216	0.047*
	Precision	0.060	0.008*
	Recall	0.442	0.675
	F1-score	0.185	0.016*
	ROC AUC	0.166	0.016*

To assess the relative predictive value of each omic modality and their combinations, we evaluated the proposed GATConvGNN model branch alone using all possible combinations of the available omic data. An "*" denotes the single omics choices considered in the multi-omics combinations below.

TABLE X
GATCONVGNN PERFORMANCE BASED ON THE OMIC CONSIDERED

Model	Omic Type	Accuracy ± SD
GATConvGNN	mRNA*	0.904 ± 0.007
GATConvGNN	Methylation (only Islands)	0.831 ± 0.029
GATConvGNN	Methylation (weighted Avg)*	0.865 ± 0.025
GATConvGNN	CNV (single value)	0.635 ± 0.022
GATConvGNN	CNV (with Min and Max)	0.625 ± 0.038
GATConvGNN	CNV (with Max)	0.627 ± 0.035
GATConvGNN	CNV (with MinMaxDiff)*	0.645 ± 0.045
GATConvGNN	mRNA + CNV	0.892 ± 0.025
GATConvGNN	Methylation + CNV	0.797 ± 0.054
GATConvGNN	mRNA + Methylation	0.934 ± 0.009
GATConvGNN	mRNA + CNV + Methylation	0.935 ± 0.018

The results show clear differences among individual modalities, with transcriptomic and methylation data generally providing stronger predictive signals compared to copy number variation (CNV), which appears less informative when used alone. When combining multiple omic layers, performance generally improves, particularly when RNA expression and DNA methylation are used together. In contrast, configurations including CNV show more limited or inconsistent gains. Overall, the best performance is achieved when all omic modalities are integrated, although the improvement over the best dual-omic combination is marginal.

B. Interpretability

To allow the interpretation of the decision process of our models, we implemented a function to compute the gradient-based saliency of the genes: it quantifies the contribution of each input node to the final prediction by retrieving its gradient for each patient in the test set, computing the importance of a single node as the mean of the absolute gradients across the node features and then averaging the genes scores for the whole test dataset to obtain a global ranking of "genes importance". It represents how much the model relies on that specific gene's values and interactions to make the final prediction, telling us which ones the analyzed model considers discriminatory drivers. We incorporated a function to help visualize the retrieved genes' features (omics) values across the dataset, in order to establish which one might capture the focus of the model. Below, we report some examples: as can be observed from the plots, the genes ENSG00000169469 and ENSG0000086548 show limited discriminative power based on beta values alone, suggesting that they are not informative for distinguishing the two classes considering only the methylation modality. At the same time, the genes ENSG00000174332 and ENSG0000086548 show no relevant differences based on their CNV values. However, their relevance is supported by the corresponding mRNA expression level and other modalities plots, where a clearer distinction between the two classes can be observed (see Fig. 11, Fig. 12 and Fig. 13). Each dot represents an individual patient sample, highlighting the expression variance across the two lung cancer classes.

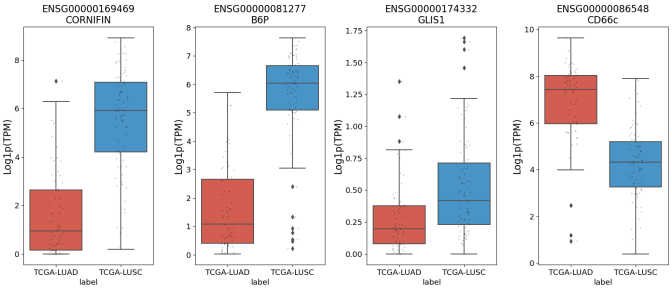

Fig. 11. Boxplots showing the distribution of TPM levels for the top 4 genes identified by the MultiModalGNN architecture.

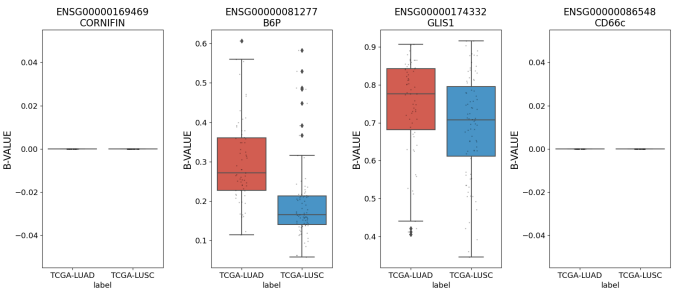

Fig. 12. Boxplots showing the distribution of beta values for the top 4 genes identified by the MultiModalGNN architecture.

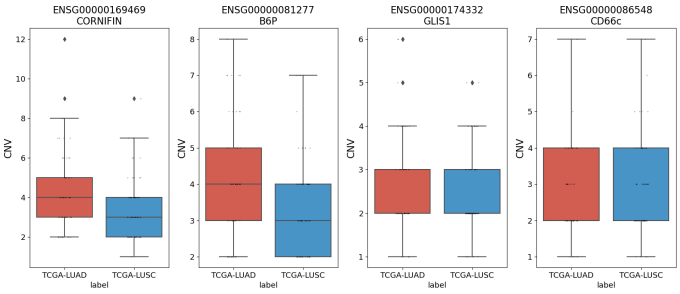

Fig. 13. Boxplots showing the distribution of copy number variation levels for the top 4 genes identified by the MultiModalGNN architecture.

Other important genes are reasonably given an high saliency, for example *KRT5* (ENSG00000186081) and *KRT13* (ENSG00000171401): those are protein coding genes typically associated with Squamous Carcinoma (LUSC) [16].

We also implemented a function to retrieve the attention scores given by the GAT attention heads to the genes; we report a few examples of genes that received attention score = 1.0 (maximum):

- *KRT17* (ENSG00000128422) and *ELFN2* (ENSG00000166897) – those two genes are established biomarkers in the distinction between LUAD and LUSC [16].
- *ALK1* (ENSG00000124107) and *DDR2* (ENSG00000162733) – other genes mentioned as related to the distinction of the two cancer classes [17]

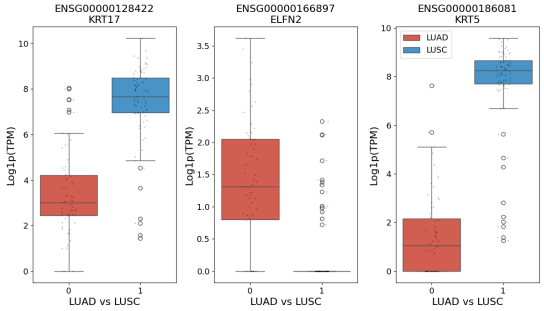

Fig. 14. Expression values of *KRT17*, *ELFN2* and *KRT5* across the two classes.

C. Performances on another dataset

To further evaluate the generalizability of the proposed architecture, we applied the same pipeline to another TCGA dataset downloaded from the GDC Data Portal [4], containing files related to kidney tumors. In this case, the classification task was more challenging than the LUAD/LUSC scenario, since the dataset involved three different tumor classes instead of a binary classification problem. The dataset consisted of 448 patients affected by either Kidney Renal Clear Cell Carcinoma (KIRC, 228 patients), Kidney Renal Papillary Cell Carcinoma (KIRP, 211 patients) or Kidney Chromophobe Carcinoma (KICH, 9 patients). Unlike the LUAD/LUSC dataset, the kidney dataset was imbalanced due to the limited number of KICH samples. The preprocessing phase remained consistent with the one adopted for the lung cancer dataset. For each patient, RNA gene expression, Copy Number Variation (CNV) and DNA methylation data were integrated to construct biological graphs based on protein–protein interactions (PPIs) retrieved from the STRING database. Node and edge features were generated using the same strategy previously described. From an architectural perspective, only the final classification layer was modified: while the LUAD/LUSC task was formulated as a binary classification problem, the kidney tumor dataset required a multi-class setting. Accordingly, we made the classifier able to dinamically reconfigure itself and produce three output scores, each corresponding to one of the tumor classes.

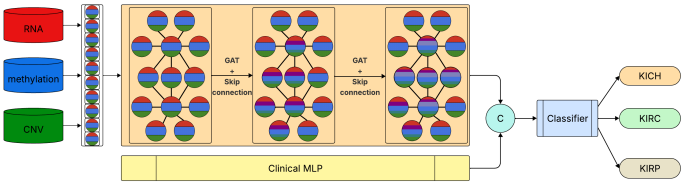

Fig. 15. Architecture adopted for the kidney tumor classification task. The model follows the same pipeline used for LUAD/LUSC classification, except for the final classification layer, which was extended to support multi-class predictions.

For the kidney dataset, the same 5-fold Stratified Cross-Validation strategy was used. The results are reported for each fold to evaluate the model performance. For this dataset, many clinical features were missing or not feasable, so their impact was found to be even less relevant.

TABLE XI
K-FOLD PERFORMANCE ON THE KIDNEY DATASET (MEAN ± STD)

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
GATConvGNN	0.88 ± 0.02	0.62 ± 0.06	0.63 ± 0.08	0.62 ± 0.07	0.97 ± 0.02
MultiModalGNN	0.89 ± 0.07	0.60 ± 0.04	0.60 ± 0.05	0.60 ± 0.05	0.94 ± 0.04

The imbalance of the dataset affects the model's behavior during training, making it biased toward the majority classes. In particular, the KICH class, which has a very limited number of samples, is the most affected by this issue. As a consequence, precision and recall are way lower than in the LUAD/LUSC experiments. The model tends to produce more false positives and false negatives, especially when trying to correctly identify the minority class, which leads to a reduction in both precision and recall values and, in turn, in the F1-score. Despite these limitations, the accuracy remains high across all folds, indicating that the model is still able to correctly classify the patients. In addition, the ROC AUC stays consistently high, suggesting that the model maintains a strong ability to distinguish between the different tumor classes, even if the predictions are less stable due to the imbalance.

By computing the importance of the clinical features by permutation, we tried to find which ones the MultiModalGNN

observes the most:

```
Clinical Features importance:
tumor_grade: 0.0556
tobacco_smoking_status: 0.0333
age_at_index: 0.0000
...
```

We also computed separately for each fold the true positives, false positives, false negatives, and true negatives in order to construct the confusion matrix for the multi-class kidney tumor classification task.

TABLE XII
CONFUSION MATRIX K-FOLD GATCONVGNN (KIDNEY DATASET)

		Predicted		
		KICH (0)	KIRC (1)	KIRP (2)
Real	KICH (0)	0.100	0.000	0.900
	KIRC (1)	0.004	0.940	0.052
	KIRP (2)	0.009	0.140	0.850

The confusion matrix shows good performance on the classes KIRC and KIRP, which are mostly correctly classified. In contrast, KICH is frequently misclassified, reflecting its limited representation in the dataset. Overall, the same effect can be observed, where the strong class imbalance negatively impacts the model's ability to correctly predict the minority class.

V. CONCLUSION

This paper presents a multimodal deep learning architecture designed to distinguish between Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC) by integrating RNA expression, DNA methylation, copy number variation (CNV) and clinical data through graph-based models and a Multi-Layer Perceptron. Patients are modeled as graphs in which nodes represent genes and edges represent functional relationships between them. Graph Attention Networks (GAT) are used to integrate multi-omics information and learn meaningful representations of the underlying biological structure. The resulting graph embeddings are then concatenated with the embeddings produced by the MLP processing the clinical features, forming a unified MultiModalGNN architecture capable of capturing complex biological signatures for the classification task.

Experimental results demonstrated that architectures implementing GATConv layers achieved the best overall performance among the evaluated models, confirming the effectiveness of graph-based approaches for representing biological interaction networks and integrating heterogeneous omics data. Moreover, the proposed framework was successfully adapted to a different tumor domain involving kidney cancer subtypes, showing good generalization capabilities even in the presence of highly imbalanced data and limited clinical information.

Overall, the proposed approach demonstrates that combining multi-omics data with clinical information can effectively discriminate between different cancer subtypes. Such models may support oncologists in improving diagnostic accuracy and potentially guiding more appropriate therapeutic strategies for patients affected by these tumor types.

REFERENCES

[1] Caterina Belluti, Sara Gherardi, "GitHub Tumor Type Classification." https://github.com/catebell/Tumor-Type-Classification.git.

[2] X. Zhang *et al.*, "A multimodal graph neural network framework for cancer molecular subtype classification." https://link.springer.com/article/10.1186/s12859-023-05622-4.

[3] Y.-C. Lin *et al.*, "Moagnn: a hierarchical graph neural network for multi-omics integration in luad subtype classification and prognosis prediction." https://academic.oup.com/bib/article/27/1/bbaf735/8429889.

[4] National Cancer Institute, "GDC data portal." https://portal.gdc.cancer.gov.

[5] STRING Consortium 2025, "String." https://string-db.org/cgi/download.pl.

[6] Chunlei Wu, Cyrus Afrasiabi, Sebastien Lelong, "mygene." https://pypi.org/project/mygene/.

[7] Illumina's Corporate, "Infinium humanmethylation450k v1.2 product files." https://support.illumina.com/downloads/infinium_humanmethylation450_product_files.html.

[8] RNA-Seq Blog, "Rpkm, fpkm and tpm, clearly explained." https://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/.

[9] Life Epigenetics, "Introduction to dna methylation analysis." https://life-epigenetics-methylprep.readthedocs-hosted.com/en/feature-v1.6.1/docs/introduction/introduction.html.

[10] PyTorch Geometric Team, "Mlp documentation." https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.models.MLP.html.

[11] PyTorch Geometric Team, "Gcnconv documentation." https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GCNConv.html.

[12] PyTorch Geometric Team, "Gineconv documentation." https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GINEConv.html.

[13] Cheng-Pei Lin, Yann-Jen Ho, Yen-Peng Chiu, Yun Tang, You Sheng Paik, Guan-Ting Chen, Wei-Chih Huang, Tzong-Yi Lee, "Moagnn: Hierarchical multi-omics graph neural network with sagpooling." https://github.com/leebiomicslab/MoAGNN.git.

[14] PyTorch Geometric Team, "Graphconv documentation." https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GraphConv.html.

[15] PyTorch Geometric Team, "Gatv2conv documentation." https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GATv2Conv.html.

[16] Chen, Joe W. and Dhahbi, Joseph, "Lung adenocarcinoma and lung squamous cell carcinoma cancer classification, biomarker identification, and gene expression analysis using overlapping feature selection methods." https://www.nature.com/articles/s41598-021-92725-8.

[17] Shen Y, Chen JQ, Li XP, "Differences between lung adenocarcinoma and lung squamous cell carcinoma: Driver genes, therapeutic targets, and clinical efficacy." https://pmc.ncbi.nlm.nih.gov/articles/PMC11904499/.

DATA AVAILABILITY

The datasets and files used in this study are publicly available from the following sources:

- TCGA multi-omics data can be downloaded from the GDC Data Portal: https://portal.gdc.cancer.gov/. Data were retrieved from the TCGA-LUAD, TCGA-LUSC, TGCA-KICH, TGCA-KIRC and TGCA-KIRP projects.
- STRING database (v12.0): https://string-db.org/cgi/download
- Illumina HumanMethylation450 manifest: https://webdata.illumina.com/downloads/productfiles/humanmethylation450/humanmethylation450_15017482_v1-2.csv