Pathway-Capsule Neural Networks

Phuong-Nam Nguyen

Faculty of Computer Science PHENIKAA University, Yen Nghia, Ha Dong Hanoi, 12116, Vietnam.

Correspondence: Phuong-Nam Nguyen, nam.nguyenphuong@phenikaa-uni.edu.vn

Received: 15/2/2024, revised: 07/5//2024, accepted: 11/5/2024

Digital Object Identifier: 10.32913/mic-ict-research-vn.v2024.n1.1265

Abstract: The progress made in Next-Generation Sequencing technology has resulted in many genetic databases, which offer the potential for more effective diagnostic tools. Recent literature highlights Artificial Intelligence (AI) as one of the most potent methods. Nonetheless, when it comes to analyzing genetic data, Machine Learning (ML) and Deep Learning (DL) face three significant challenges due to: (1) multi-modality analysis, (2) model explainability, and (3) extension to patient-specific biomarker identification. In this paper, we introduce Pathway-Capsule Neural Networks (*PCNN*) and its Bayesian version (*BayesPCNN*) to address the three limitations. We demonstrate the model using Big Data of $7,372$ samples with 57 protein expressions from the TCGA dataset. In the best setting, the proposed PCNN outperforms tree-based ML models and KNN with a significant gap of $10\% - 30\%$ in accuracy for Pan-Cancer detection; while only outperformed by kernel methods with under 2%. Another contribution of BayesPCNN is highlighted as Explainable AI (*XAI*), which reveals tumors with potential metastasis and patient-specific biomarkers.

Keywords: *deep learning, artificial intelligence, bioinformatics*

Tiêu đề: **Pathway-Capsule Neural Networks**

Tóm tắt: Tiến bộ trong Công nghệ Sequencing Thế hệ Tiếp theo đã dẫn đến nhiều cơ sở dữ liệu gen, mở ra tiềm năng cho các công cụ chẩn đoán hiệu quả hơn. Các báo cáo khoa học quốc tế đương đại nhấn mạnh rằng Trí tuệ Nhân tạo (AI) là một trong những phương pháp mạnh mẽ nhất. Tuy nhiên, khi đến việc phân tích dữ liệu gen, Học Máy (ML) và Học Sâu (DL) đối mặt với ba thách thức lớn do: (1) phân tích đa dạng, (2) khả năng giải thích mô hình, và (3) mở rộng để xác định chỉ số sinh học cụ thể cho từng bệnh nhân. Trong bài báo này, chúng tôi giới thiệu Pathway-Capsule Neural Networks (*PCNN*) và phiên bản Bayesian của nó (*BayesPCNN*) để giải quyết các hạn chế này. Chúng tôi thể hiện mô hình bằng Dữ liệu Lớn của $7,372$ mẫu với 57 biểu hiện protein từ tập dữ liệu TCGA. Trong cài đặt tối ưu, PCNN được đề xuất vượt trội hơn so với các mô hình ML dựa trên cây và KNN với khoảng cách đáng kể là $10\% - 30\%$ trong độ chính xác cho việc phát hiện Ung thư Phổ quát; trong khi chỉ bị vượt qua bởi các phương pháp nhân bản với dưới 2%. Một đóng góp khác của BayesPCNN được nhấn mạnh là Trí tuệ Nhân tạo Giải thích (*XAI*), làm sáng tỏ những khối u có tiềm năng di căn và chỉ số sinh học cụ thể cho từng bệnh nhân.

Từ khóa: *học sâu, trí tuệ nhân tạo, toán y sinh*

I. INTRODUCTION

The next-generation sequencing (NGS) technology advancement brings a large collection of genetic databases, which promises more powerful diagnostic tools. Artificial Intelligence (AI) is one of the most powerful approaches in the recent associated literature. However, Machine Learning (ML) and Deep Learning (DL) deal with a major challenge when analyzing genetic data. Specifically, most AI-driven solutions are black-box, which is not ideal for medical context as clinicians must understand the model inference to offer precise treatments for the *patient of target*.

In this paper, we propose a new neural architecture named *Pathway-Capsule Neural Networks* (PCNNs) to achieve three goals

1) Our proposed dual-encoder retrieves hierarchical information from multi-mode genetic data using a generalized loss module (Section III.2 and III.3).
2) The proposed AI model has two settings: (1) generic PCNN and (2) BayesPCNN, which is more accurate than most conventional ML models for genetic data. Both versions outperform four competitor models in the Pan-Cancer detection task while slightly outperformed by the Support Vector Machine (SVM) (Section IV).
3) BayesPCNN is an XAI model identifying secondary tumors and patient-specific biomarkers. (see Section IV.2.b and IV.2.c).

II. RELATED WORKS

1. Capsule Networks in Cancer Genetics

Capsule Networks, or CapsNets, are an Artificial Neural Network (ANN) that was proposed to address some of the limitations of Convolutional Neural Networks (CNNs) [1]. CapsNets solve the problem of preserving the hierarchical relationships between parts and wholes, and they have rotational and positional invariance[2]. They use a group of neurons to form a "capsule" that recognizes an object in the input data and outputs a vector. The vector's length represents the object's probability of being present in the input, and its orientation represents the object's properties, such as pose, deformation, and texture. In the context of cancer genetics data analysis, capsule neural networks are used to improve the accuracy, robustness, and the model interpretation[3–9].

$$\textbf{NNs}: \boldsymbol{X} \xrightarrow{f(\boldsymbol{\theta})} \boldsymbol{h}(\dim = 1) \longrightarrow \text{CLS}$$
$$\boldsymbol{\theta} \xleftarrow[\text{Stochastic Opt.}]{\text{update}} \mathcal{D}(\boldsymbol{X}, \boldsymbol{Y})$$

$$\textbf{CapsNet}: \boldsymbol{X} \xrightarrow{f(\boldsymbol{\theta})} \boldsymbol{h}(\dim \geq 2) \longrightarrow \text{CLS}$$
$$\boldsymbol{\theta} \xleftarrow[\text{Dynamic Routing}]{\text{update}} \mathcal{D}(\boldsymbol{X}, \boldsymbol{Y})$$

2. Challenges of AI models in Cancer Genomics

Genetic data analysis model architectures still need improvement despite many achievements. A recent review[10] shows tree-based and kernel methods are more commonly used than neural networks because Deep Neural Networks (DNN) are ineffective in learning from tabular data. DNNs are good at learning representations from data with physically structured domains, such as images or text sequences. However, tabular data does not have generic physical structures. Thus, DNN performance degrades, especially in low-limited data scenarios[11].

III. METHOD: PATHWAY-CAPSULE NEURAL NETWORK (PCNN)

1. Overview

We compare our proposed PCNN to conventional NN and CapsNets in Figure 1. Our model architecture has three distinct features:

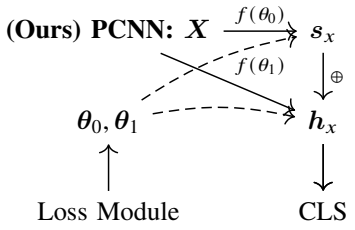

Figure 1: Pathway-Capsule Neural Networks (PCNN) Information Flow, compared to conventional NN and CapsNets.

1) We partition the model weight architecture as a capsule $\boldsymbol{s}_x$ and $\boldsymbol{h}_x$, which encodes the hierarchical structure on inputs.
2) The loss module is designed to update each cluster of model weight based on mutational signals and true labels.
3) The final classifier is produced by both secondary representation $\boldsymbol{s}_x$ and primary representation $\boldsymbol{h}_x$ though element-wise sum, which is inspired by residual connections in ResNet[12]. It allows us to use deeper model architecture with more efficient trainability.

2. Dual-Neural Encoder for Multi-Modality Analysis

We propose a novel neural architecture design for ANN models named Pathway-Capsule Neural Networks (PCNN), which leverages a dual-encoder for multi-modality analysis. Specifically, given two modes of data in cancer genetics, denoted as $D^1 = (\boldsymbol{X}_1, \boldsymbol{Y}_1)$ and $D^2 = (\boldsymbol{X}_2, \boldsymbol{Y}_2)$, in which $\boldsymbol{Y}_1$ is labels for cancer classes and $\boldsymbol{Y}_2$ is the mutational signals. The input data $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ can be any genetic data ranging from DNA methylation, RNA/mRNA, or protein expression.

Construction of Pathway-Capsule ($g(.)$): We propose a capsule-like neural architecture to encode mutational signals for input pathways. Specifically, given the input data of k-genetic features $\boldsymbol{X}_1 = \boldsymbol{X}_2 = \{G_1, \ldots, G_k\}$, the Pathway-Capsule block $g(.)$ encodes the input signals as the following forward-pass in **Algorithm** 1. The output of the Pathway-Capsule block is the neural signal that represents mutational activation signals (binary $\in \{0, 1\}$). We denote the transformation of this neural block as $g(\boldsymbol{X}) = \boldsymbol{s}_x$.

Construction of Classifiers ($f(.)$): The final ANN classifier is parameterized by ($n_{\text{capsules}} = k, n_{\text{layers}} = l, \text{hidden dim} = d$), in which n_{capsules} Pathway-Capsule block encodes mutational signals of k-genetic features with l layers of d hidden neurons in each block. We aggregate

the neural signals from all Pathway-Capsule blocks by concatenation, which is given as

$$\boldsymbol{H} = \text{Concat}(\boldsymbol{s}_x^i)_{i=1}^k. \tag{1}$$

We improve the trainability of the proposed model architecture by incorporating the original signals $\boldsymbol{X}_{(.)}$ into the embedding $\boldsymbol{H}$. Of note, residual connections[12] is already adopted in the Pathway-Capsule block for the same purpose depicted in the demonstrated forward pass. Here, we use the concatenation operator to construct such a representation, given as

$$\boldsymbol{H}_c = \text{Concat}(\boldsymbol{X}, \boldsymbol{H}). \tag{2}$$

The proposed representations $\boldsymbol{H}_c$ could have two-fold advantages. First, the input signal is accounted for in such an embedding so that it could ameliorate the model trainability. Second, the embeddings contain hierarchical information from multi-mode data, including (1) input genetic expressions and (2) mutational signals. The final classification head $f(.)$ is built on $\boldsymbol{H}_c$, which is given as

$$\hat{\boldsymbol{y}} = f(\boldsymbol{H}_c), \tag{3}$$

where $\hat{\boldsymbol{y}}$ is the predicted outcome. The forward pass of the proposed classifier is given as in **Algorithm** 2.

3. Loss Module

Our proposed PCNN model is different from Capsule Neural Networks and ANNs with auxiliary heads in [1, 3, 6, 8]. It encodes hierarchical information for multi-modality analysis and includes a loss module to learn class annotations and mutational signals. This is the key feature that sets it apart from existing models.

Here, we design the loss module consisting of two components, given as

$$\mathcal{L} = \mathcal{L}_{\text{Cls}} + \mathcal{L}_{\text{Mutation}} \tag{4}$$

Of note, the annotations for model feedback include $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$, which is introduced in the earlier section. The classification loss is computed as

$$\mathcal{L}_{\text{Cls}}(\hat{\boldsymbol{y}}, \boldsymbol{Y}_1), \tag{5}$$

using cross-entropy criterion given as

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

where N is the number of samples, M is the number of classes, y_{ij} is the binary indicator (0 or 1) if class j is the true label for sample i p_{ij} is the predicted probability that sample i belongs to class j. The latter term in the final loss is computed between neural signals $\boldsymbol{s}_x^i$ and mutational signals recorded in each k-genetic feature. Specifically, we use the binary version of cross-entropy loss, yielding

$$\mathcal{L}_{\text{Mutation}} = \sum_{i=1}^{k} \mathcal{L}_{BCE}(\boldsymbol{s}_x^i, \boldsymbol{y}_2^i), \tag{6}$$

where $\boldsymbol{y}_2^i$ is the mutational-activation signals from $\boldsymbol{Y}_2$.

IV. RESULTS

1. Experimental Design

a) Case-study

Our proposed PCNN was demonstrated using RPPA and CNV, from DNA mutation to protein expressions. We used TCGA[13] with 7,372 patients, 31 cancer types, and $k = 57$ genetic features per observation. Mutational signals were created by CNV data with simple thresholding of $\boldsymbol{Y}_2 = 1$ when $y_{\text{CNV}} \neq 0$, otherwise 0.

b) Training Protocol

We optimize hyper-parameters using Sequential Model-based Optimization with Tree-Parzen Estimator[14]. This optimizer is cost-efficient for large search spaces[15, 16]. We optimize the number of capsule layers, neurons per layer, dropout rate, learning rate, and weight decay. We perform 100 trials using SMBO-TPE[14]. Each model candidate trains for 1000 epochs using the AdamW[17] optimizer. We conduct experiments using Python 3.7.0, numpy 1.21.5, scikit-learn 1.0.2, and PyTorch 1.11 on an Intel i9 processor, 16GB DDR4 memory, and GeForce GTX 1060 Mobile GPU.

2. Effectiveness Analysis

a) PCNN Enables More Accurate ML Classifiers for Pan-Cancer Detection

We compared our proposed PCNN model with five classical ML models and excluded ANNs for two reasons. *First*, PCNN is a more generalized version of ANNs. *Second*, the ML models have demonstrated superior performance in genetic data analysis on TCGA data[10]. We used the test accuracy score evaluated by a 5-Fold CV as the comparison metric and optimized all models using SMBO-TPE for a fair comparison.

We present two versions of PCNNs in Figure 2 - one based on a generic setting and the other on a Bayesian setting (by DeepEnsemble[18]). The generic PCNN model outperforms four competitor models, except SVM with radial kernel parameterized by $C = 5$. The gap between PCNN and decision tree and KNN models is large, with a difference of approximately 92.5% compared to 62.5% and 82.5%, respectively. PCNN also performs slightly better than other tree-based approaches by about 2.5% – 5% in test accuracy.

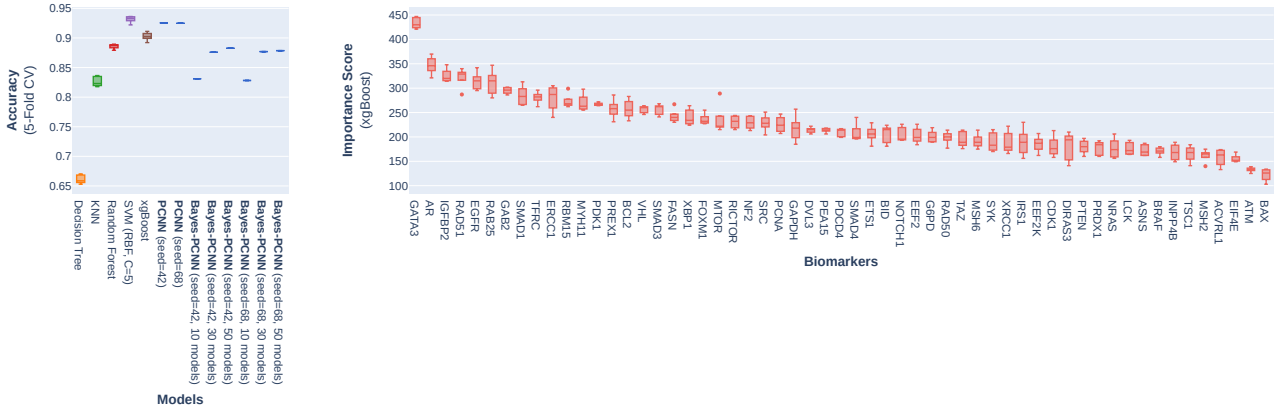

Figure 2: **(Left)** Our proposed generic PCNN and Bayes-PCNN outperform competed for tree-based classifiers, including decision tree, KNN, and xgBoost, while slightly outperformed by SVM using 5-Fold Cross-Validation (CV). **(Right)**: Proteomics-Biomarker scored by importance score from xgBoost models based on 5-Fold CV.

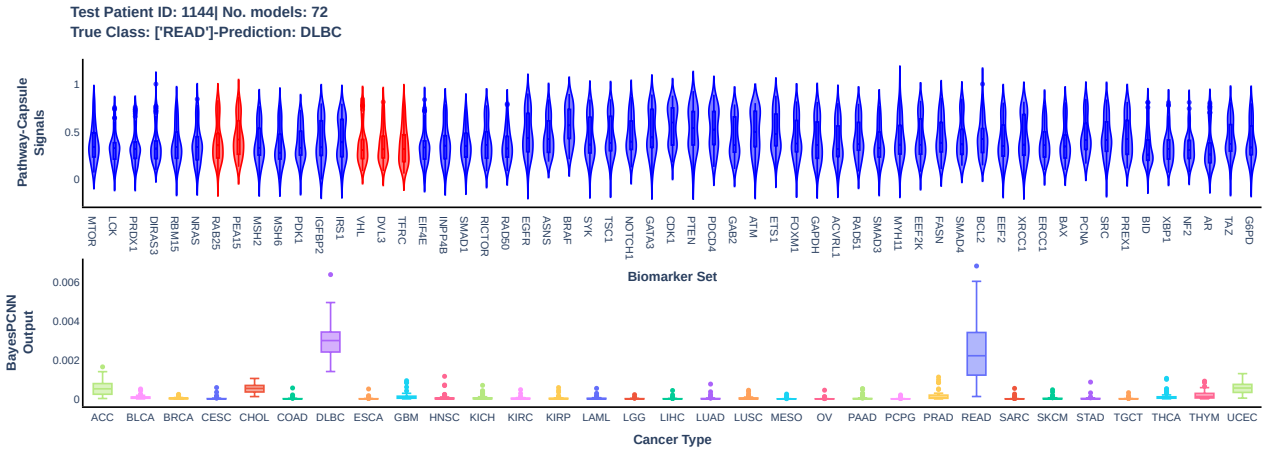

Figure 3: **Bayes-PCNN Improves Patient-specific Inference. (Top)** Neural signals recorded in Pathway-Capsule outputs, colored by mutational signal: red for mutation and blue for wild-type. **(Bottom)** Assembled predictions by Bayes-PCNN using 72 optimized models: Although the top-1 prediction is wrong, the second highest confidence is the right cancer class.

The predictive performance of PCNN is slightly lower than SVM by about 2.5%.

BayesPCNN, a variant of PCNN in the Bayesian setting, achieves lower accuracy than the generic version due to its aggregation of top-10, -30, and -50 to produce the mean of predicted values. While more classifiers result in higher accuracy, there is a trade-off between inference time and accuracy. However, BayesPCNN using ten to thirty models gains a 10% accuracy gain, which saturates at 50 models, whereas the additional 20 classifiers only gain $< 2\%$ accuracy.

b) Enhancing The Robustness of Model Inference with Bayes-PCNN

Although Bayes-PCNN induces a slight degradation in accuracy, the model setting has two-fold advantages. *First*, the model produces a more robust and reliable inference thanks to the nature of Bayesian learning. We show in Figure 3 that even if the classifier makes the wrong prediction in its top-1 configuration, the second highest confident prediction is the right cancer class. *Second*, we can associate top-2 and top-3 confidence prediction as potential metastasized diseases (secondary). The analyzed sample contains genetic signals that resemble other cancer classes.

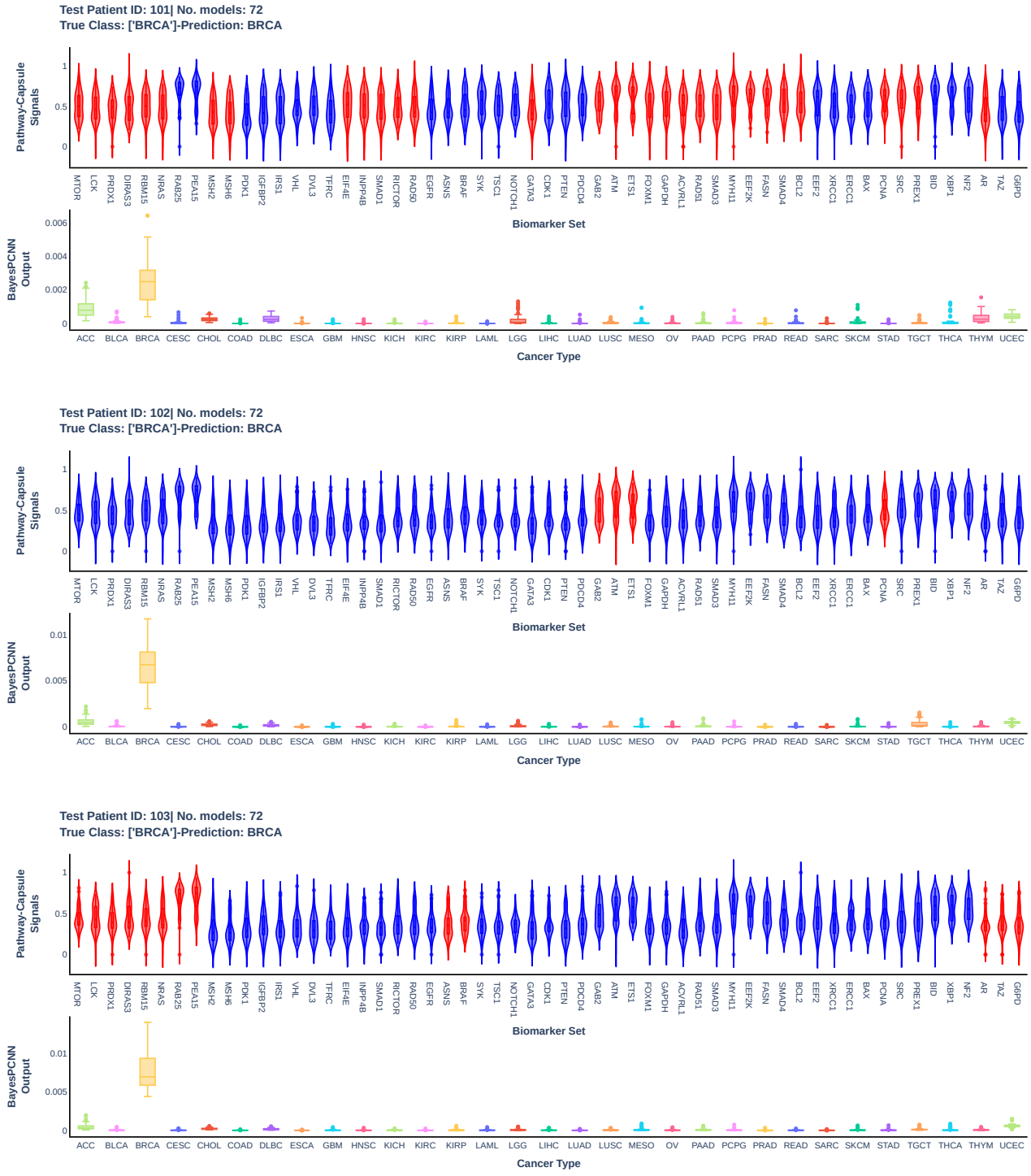

Figure 4: **Explainable AI (XAI) by Bayes-PCNN Demonstrated on Comparing Different Patients with *Breast Cancer*.** The top panel depicts neural signals recorded from the output of Pathway-Capsules blocks, colored by true mutational signals (red for mutant, blue for wild-type). The bottom panel depicts the confidence in model inference by cancer diseases. The predicted metastasized tumor has *MSH2*, *MSH6*, *EIF4E*, *INPP4B* and *SMAD1* mutations, that associates with high values in capsule signals

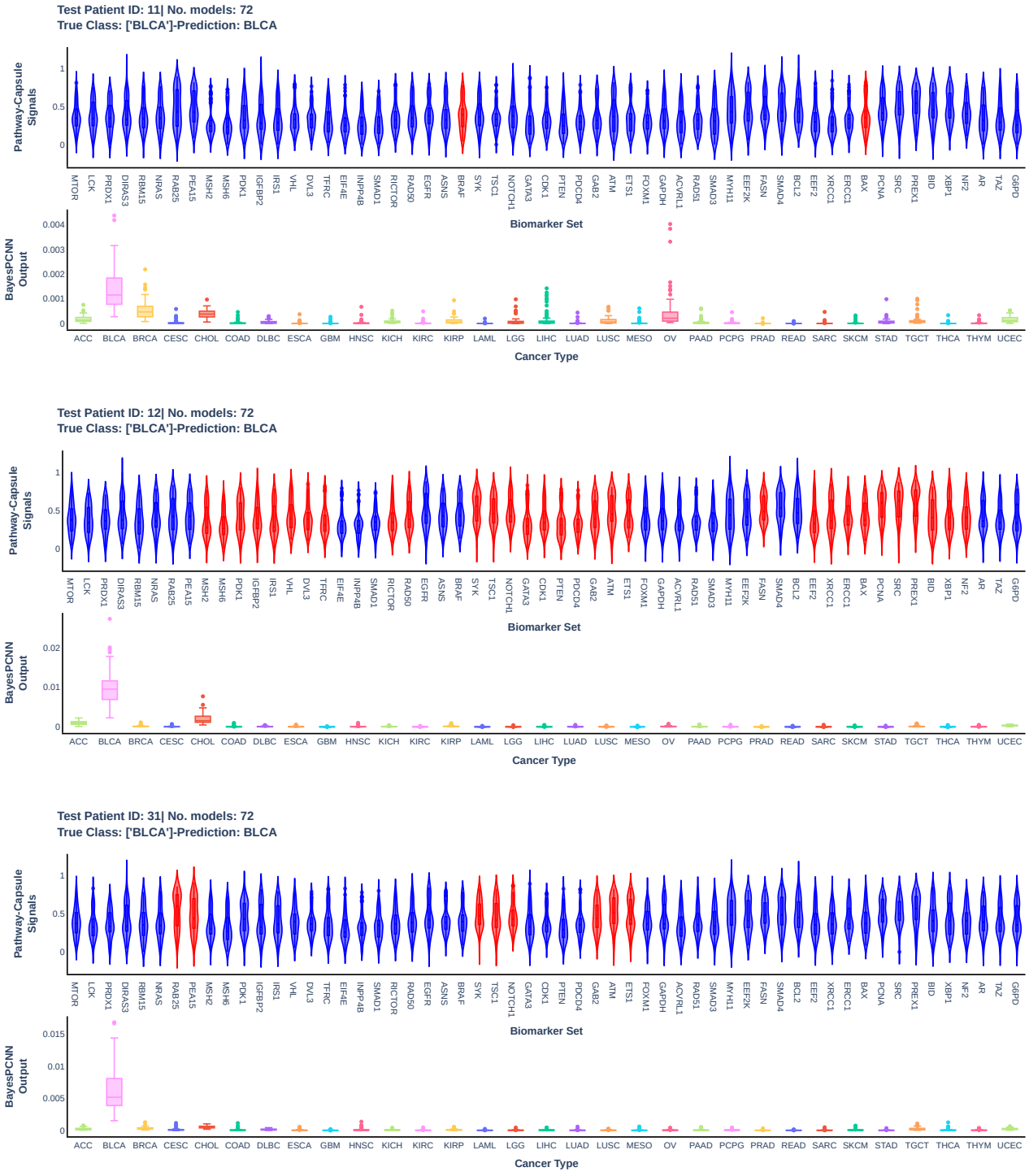

Figure 5: **Explainable AI (XAI) by Bayes-PCNN Demonstrated on Comparing Different Patients with *Bladder Cancer*.** The top panel depicts neural signals recorded from the output of Pathway-Capsules blocks, colored by true mutational signals (red for mutant, blue for wild-type). The bottom panel depicts the confidence in model inference by cancer diseases. The co-activation of *BRAF* and *BAX* could lead to potential metastasis to secondary at the breast, suggested by our BayesPCNN.

For instance, BayesPCNN suggests the diagnosed patient in Figure 3 could have potential disease metastasis from the rectum (READ) to Lymphoid Neoplasm Diffuse Large B-cell Lymphoma (DLBC) or adrenocortical carcinoma (ACC). The two discussed features are superior to the existing models, which are considered black-box.

c) PCNN Enables Explainable-AI (XAI) for Patient-specific Diagnostics

We used Pathway-Capsule blocks to understand model inference and identify clinically significant observations in breast cancer patients. Figure 4 shows the Pathway-Capsule output, colored according to mutational signals. Our XAI system identified patient-specific biomarkers, whereas existing scoring systems for feature importance only improve model accuracy.

Patient 101 may have potential metastasis to ACC, while the other patients are not at risk for secondary tumors. The predicted metastasized tumor has mutations in *MSH2*, *MSH6*, *EIF4E*, *INPP4B*, and *SMAD1*, which are associated with high values in capsule signals. *MSH2* and *MSH6* staining absence in tumors can be due to genetic mutations inherited from the germline[19], and the risk of having *MSH2* microsatellite instability (MSI) is higher in males with colon tumors, late-stage tumors, and poor tumor differentiation. Similarly, males with colon tumors and tumors at a late stage are more likely to have *MSH6* MSI. *SMAD1* promotes cell invasion and metastasis in different cancer types[20].

We can see three bladder tumor patients in Figure 5. Patient 12 has the most mutations, but our Bayes-PCNN model shows that patient 11 has a higher risk of metastasis due to the co-activation of *BRAF* and *BAX* genes. *BRAF* mutations[21] can cause carcinogenesis[22] and are linked to poor prognosis in 5-10% of metastatic colorectal cancers. Activation of the *BAX* driver gene triggers apoptosis, leading to the death of cancer cells.

We found that neural signals from Pathway-Capsule and mutational activation can predict disease metastasis risk. For personalized treatment, we identified potential proteomics biomarkers like *SMAD1* and *BAX*.

V. DISCUSSION

Why PCNN is More Generalized than Conventional ANN Design in the Context of Multi-Modality Analysis?

The proposed PCNN is superior to conventional machine learning models as it encodes information from multiple genomic data modes into the final latent vectors. Multi-modalities analysis is crucial in AI-driven genetic analyses, but the high-dimensional data can impact the effectiveness and efficiency of machine/deep learning models. Our proposed PCNN builds on recent proposals for the path towards General AI (GAI). For instance, GLOM[2] suggests representing part-whole hierarchies in neural networks, which is achieved in the proposed PCNN by the dual-encoder. Additionally, our proposal adopts the trainable cost module[23]. However, the intrinsic cost module still needs addressing. Here, we design this module to encode mutational signals from the fundamental mode of the DNA codes.

Some Limitations of The Model Architecture

We have identified some limitations with our proposed neural architecture design. Firstly, our model requires additional costs for data annotations of mutational signals, which can be significant in medical data. However, this issue can be addressed by high-end pre-training protocols such as self-supervised learning. *Secondly*, we are also concerned about the model's scalability, as our architecture consists of dense auxiliary capsules, which could lead to high latency in model training and inference.

VI. CONCLUSION

To this end, I have introduced a novel architecture PCNN with a dual-neural-encoder to retrieve information from multi-modalities of genetic data in Section III. Although the model architecture can be generalized on various data scenarios, I have demonstrated the proof of concept using Big Data, including mutational signals and protein expressions in Section IV.1.a. The numerical experiment yields promising results three-fold. *First*, the generic PCNN outperformed four in five classical ML competitor models (Section IV.2.a). *Second*, the BayesPCNN improves the robustness of model inference and suggests a potential secondary site of tumor based on data-driven inference (Section IV.2.b). *Finally*, the final XAI-driven analysis yields robust proteomics-biomarkers demonstrated in some cancer types (Section IV.2.c). Besides, I also highlighted the motivation and explained why our proposed PCNN guarantees the improvements in Section V with the model limitation is discussed in Section V. For further study, I suggest several directions, such as investigating the proposed PCNN and BayesPCNN in other learning tasks not restricted to genetic data. Besides, in-depth studies of model interpretation to discover new biomarkers are worth investigating. Finally, addressing the discussed limitation could improve the effectiveness and scalability of the proposed PCNN.

REFERENCES

[1] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[2] G. Hinton, "How to represent part-whole hierarchies in a neural network," *Neural Computation*, pp. 1–40, 2022.
[3] C. Peng, Y. Zheng, and D.-S. Huang, "Capsule network based modeling of multi-omics data for discovery of breast cancer-related genes," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 17, no. 5, pp. 1605–1612, 2019.
[4] P. Zheng, Y. Qi, X. Li, Y. Liu, Y. Yao, and G. Huang, "A capsule network-based method for identifying transcription factors," *Frontiers in Microbiology*, vol. 13, p. 1048478, 2022.
[5] Y. Wang, L. Huang, S. Jiang, Y. Wang, J. Zou, H. Fu, and S. Yang, "Capsule networks showed excellent performance in the classification of herg blockers/nonblockers," *Frontiers in pharmacology*, vol. 10, p. 1631, 2020.
[6] L. Wang, X. Miao, R. Nie, Z. Zhang, J. Zhang, and J. Cai, "Multicapsnet: a general framework for data integration and interpretable classification," *Frontiers in genetics*, p. 2237, 2022.
[7] X. Luo, X. Kang, and A. Schönhuth, "Predicting the prevalence of complex genetic diseases from individual genotype profiles using capsule networks," *Nature Machine Intelligence*, vol. 5, no. 2, pp. 114–125, 2023.
[8] L. Wang, R. Nie, Z. Yu, R. Xin, C. Zheng, Z. Zhang, J. Zhang, and J. Cai, "An interpretable deep-learning architecture of capsule networks for identifying cell-type gene expression programs from single-cell rna-sequencing data," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 693–703, 2020.
[9] G. Hadash, E. Kermany, B. Carmeli, O. Lavi, G. Kour, and A. Jacovi, "Estimate and replace: A novel approach to integrating deep neural networks with existing applications," *arXiv preprint arXiv:1804.09028*, 2018.
[10] J. Liñares-Blanco, A. Pazos, and C. Fernandez-Lozano, "Machine learning analysis of tcga cancer data," *PeerJ Computer Science*, vol. 7, p. e584, 2021.
[11] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.
[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[13] D. C. C. B. R. . J. M. A. . K. A. . P. T. . P. D. . W. Y. 53 and T. S. S. L. D. A. 68, "The cancer genome atlas pan-cancer analysis project," *Nature genetics*, vol. 45, no. 10, pp. 1113–1120, 2013.
[14] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.
[15] N. Nguyen and K.-C. Chen, "Quantum embedding search for quantum machine learning," *IEEE Access*, vol. 10, pp. 41 444–41 456, 2022.
[16] N. Nguyen and J. M. Chang, "Csnas: Contrastive self-supervised learning neural architecture search via sequential model-based optimization," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 4, pp. 609–624, 2021.
[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
[18] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.
[19] C. D. Putnam, R. D. Kolodner *et al.*, "Dna mismatch repair: Mechanisms and cancer genetics," 2019.
[20] M.-J. Nokin, J. Bellier, F. Durieux, O. Peulen, G. Rademaker, M. Gabriel, C. Monseur, B. Charloteaux, L. Verbeke, S. van Laere *et al.*, "Methylglyoxal, a glycolysis metabolite, triggers metastasis through mek/erk/smad1 pathway activation in breast cancer," *Breast Cancer Research*, vol. 21, no. 1, pp. 1–19, 2019.
[21] T. Xu, J. Li, Z. Wang, X. Zhang, J. Zhou, Z. Lu, L. Shen, and X. Wang, "Real-world treatment and outcomes of patients with metastatic braf mutant colorectal cancer," *Cancer Medicine*, 2023.
[22] P.-P. Wang, C. Lin, J. Wang, G. A. Margonis, and B. Wu, "Braf mutations in colorectal liver metastases: Prognostic implications and potential therapeutic strategies," *Cancers*, vol. 14, no. 17, p. 4067, 2022.
[23] Y. LeCun, "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27," *Open Review*, vol. 62, 2022.

APPENDIX

Implementation

```
class PathwayCapsule(nn.Module):
  def __init__(self, in_dim, hid_dim,
    n_cap_layers, dropout):
    super(PathwayCapsule, self).__init__()
    self.dropout = dropout
    self.fc_in = nn.Linear(in_dim, hid_dim)
    self.layers = nn.ModuleList([nn.Linear(
    hid_dim, hid_dim)
                  for _ in range(
    n_cap_layers)])
    self.fc_out = nn.Linear(hid_dim, 1)
  def forward(self,x):
    x = self.fc_in(x)
    for f in self.layers:
      x = f(x) + x
      x = nn.Dropout(self.dropout)(x)
      x = nn.ReLU()(x)
    x = self.fc_out(x)
    return nn.Sigmoid()(x)
```

Listing 1: **Forward-Pass of Pathway-Capsule**

```
class Model(nn.Module):
  def __init__(self, in_dim, hid_dim, n_caps
    , n_cap_layers, out_dim, dropout = 0.):
    super(Model, self).__init__()
    self.in_dim, self.n_caps = in_dim,
    n_caps
    self.dropout = dropout
    self.capsules = nn.ModuleList([
    PathwayCapsule(in_dim, hid_dim,
    n_cap_layers,dropout) for _ in range(
    n_caps)])
    self.cls_head = nn.Linear(in_dim +
    n_caps, out_dim, bias = False)
  def forward(self, x):
    x = x.float()
    sx = []
    for f in self.capsules:
      sx.append(f(x))
    H = torch.cat(sx, dim = 1)
    Hc = torch.cat([x,H], axis = 1)
    y = self.cls_head(Hc)
    return H, y
```

Listing 2: **Forward-Pass of PCNN.**

Phuong-Nam Nguyen is a current Ph.D. student in Electrical Engineering at the University of South Florida. He earned a Bachelor's degree in Mathematics and Education from the Hanoi University of Education and a Master's in Statistics from the University of South Florida. He is currently a lecturer and research fellow at Phenikaa University. His research interests are applying classical and quantum machine intelligence to biological problems, including oncology research, molecular dynamics modeling, large genome data analysis, and biomarker identification.
Email: nam.nguyenphuong@phenikaa-uni.edu.vn