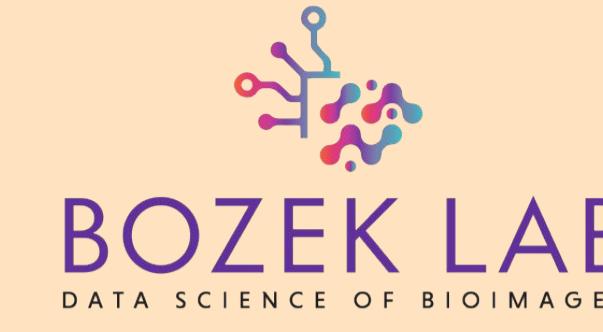


# Context-aware skin cancer cell classification based on GNNs

Lucas Sancré<sup>1,2,3</sup>, Katarzyna Bozek<sup>2,3,4</sup>



<sup>1</sup> Faculty of Mathematics and Natural Sciences, University of Cologne, Cologne, NRW, Germany

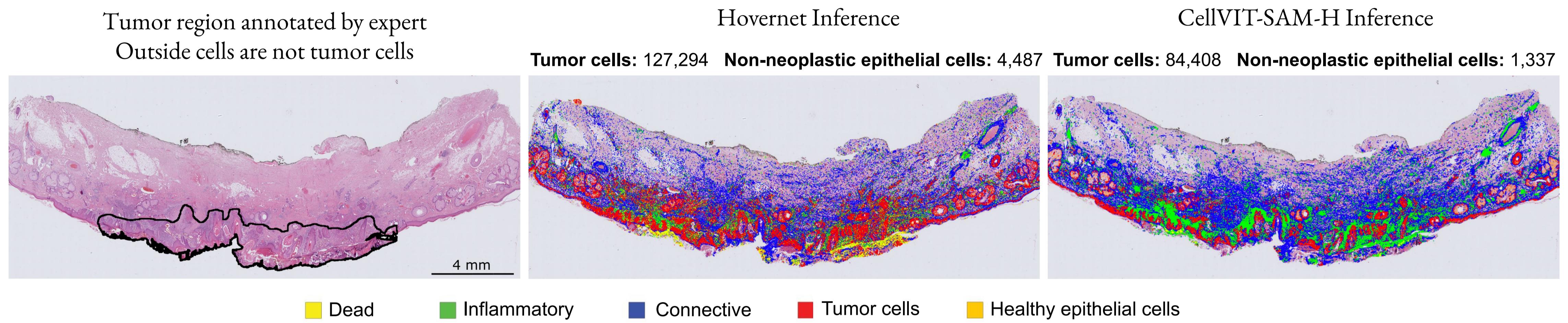
<sup>2</sup> Institute for Biomedical Informatics, Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, NRW, Germany

<sup>3</sup> Center for Molecular Medicine Cologne (CMMC), Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, NRW, Germany

<sup>4</sup> Excellence Cluster on Cellular Stress Responses in Aging-Associated Diseases (CECAD), University of Cologne, Cologne, NRW, Germany

## CONTEXT

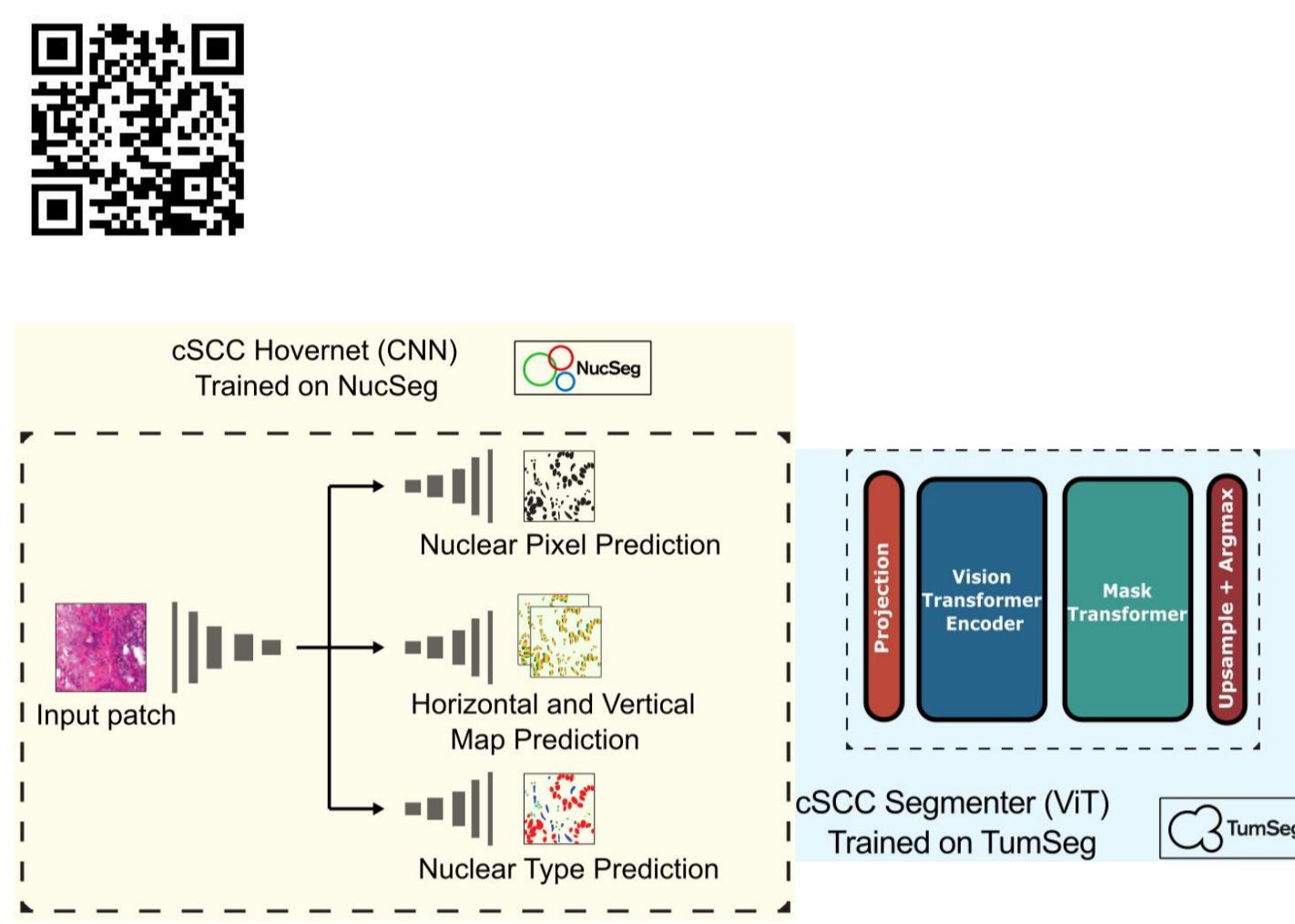
Healthy and tumoric epithelial cells are poorly discriminated by SOTA computer vision models trained for cell classification



## PREVIOUS METHOD

**Histo-Miner:** use of 2 models: one at cellular-level, for cell classification, one at tissue level for tumor segmentation

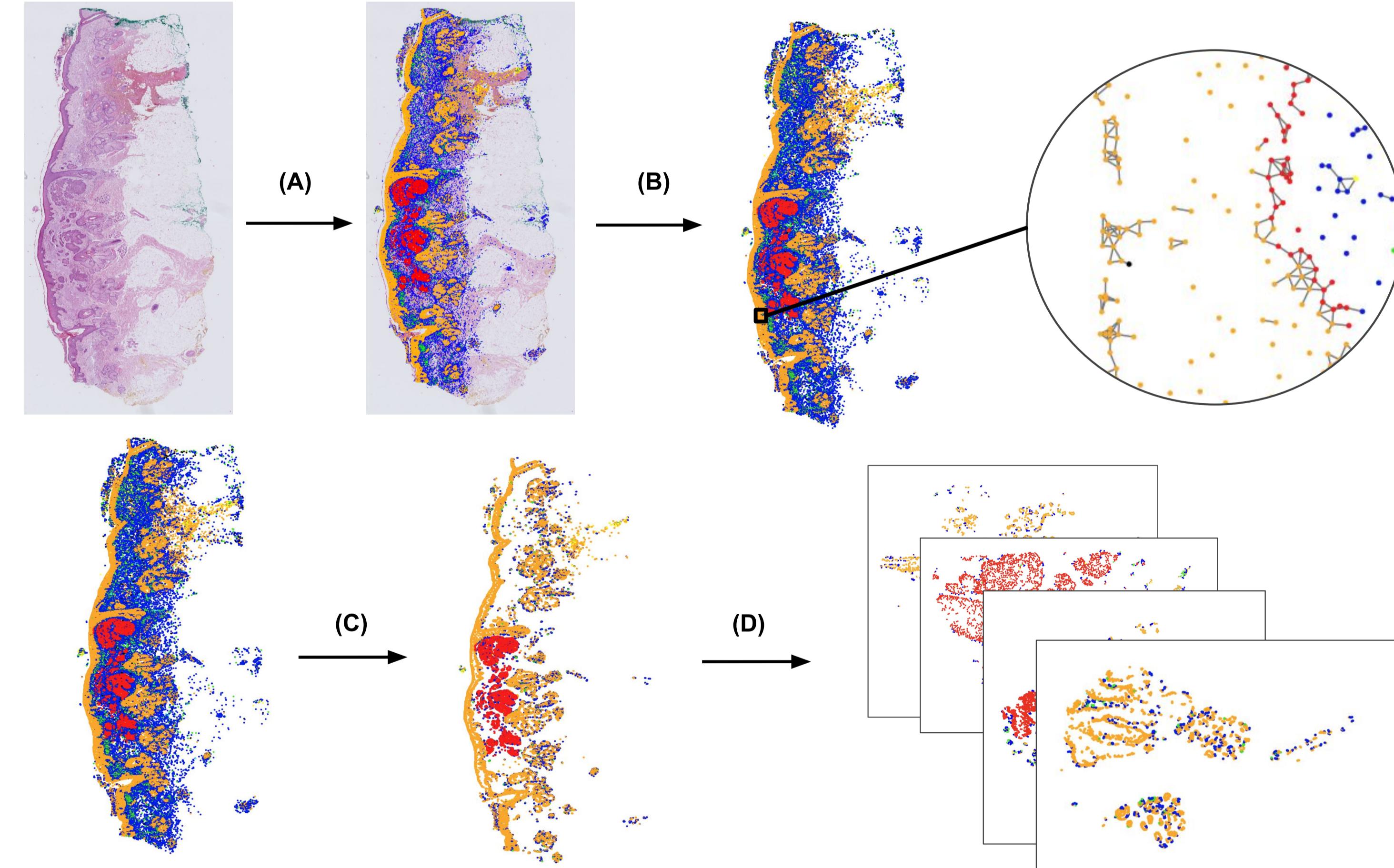
Works better than SOTA but can miss isolated tumor and healthy epithelial cell



## GENERATING GRAPH DATASET

We generate a graph dataset from a Megapixel H&E image. We train for node classification instead of cell segmentation and classification

Nodes contain: morphology + texture features and cell class information



(A) Generate SCC Hovernet computer vision inference on tumor annotated dataset

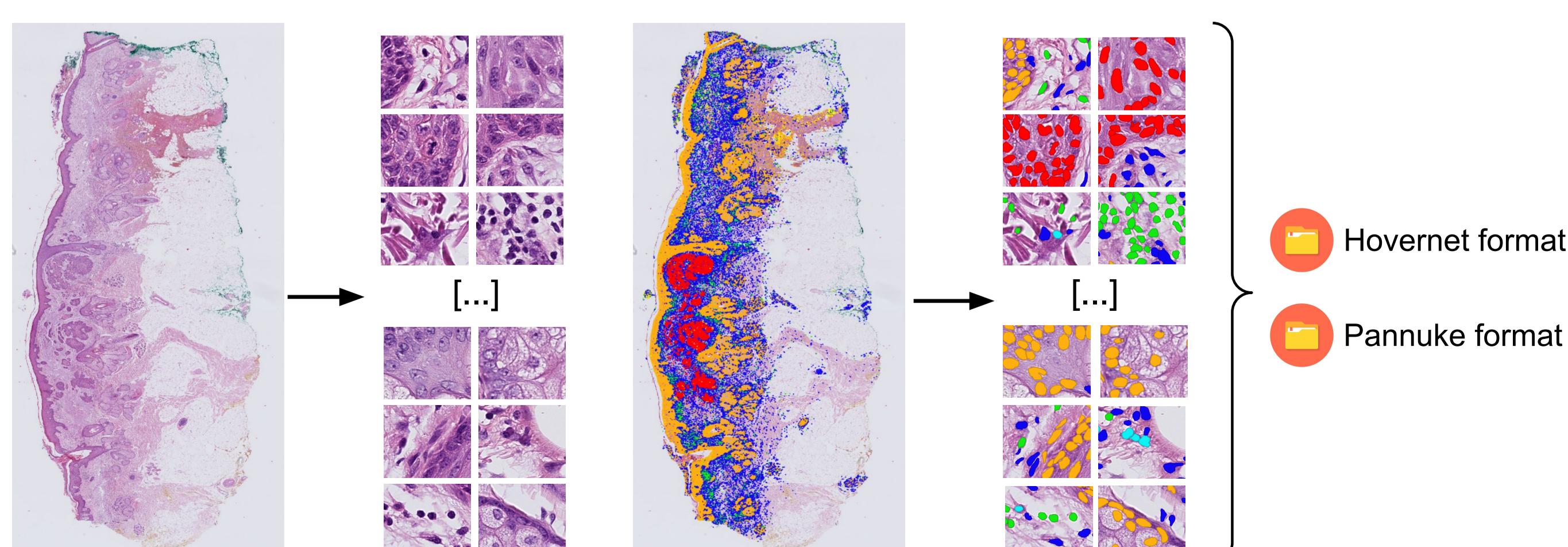
(B) Create graph from prediction: each cell is a node. Close cells (setting to set) are link with edges

(C) Graph simplification: remove unknown class nodes and limit max-hops

(D) Graph splitting: big graph not fitting in VRAM; split into batches following spatial kmeans groups

## GENERATING BASELINE DATASET

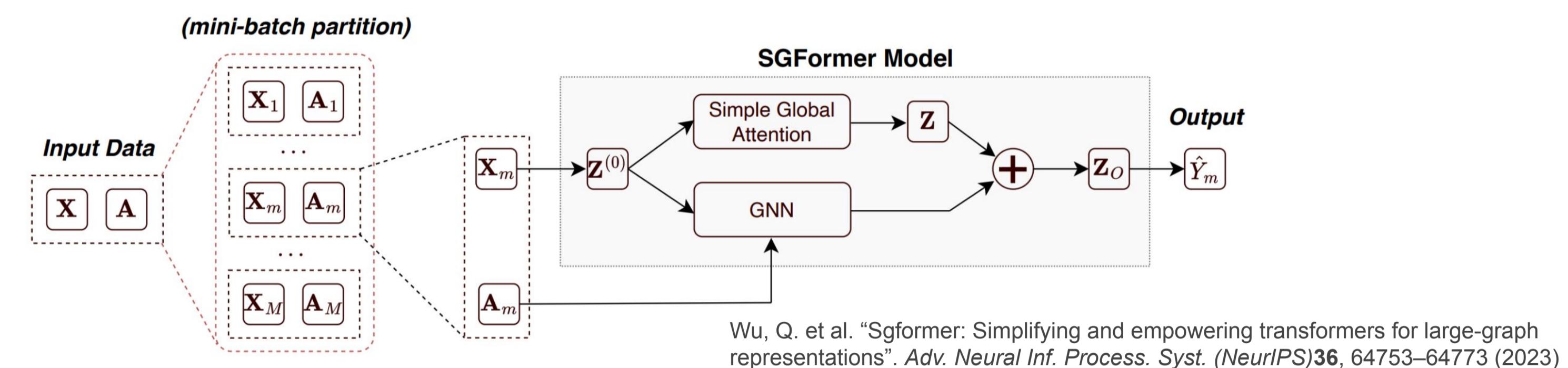
Create similar dataset to train computer vision model on classifying healthy and tumor epithelial cells



## METHOD & PRELIMINARY RESULTS

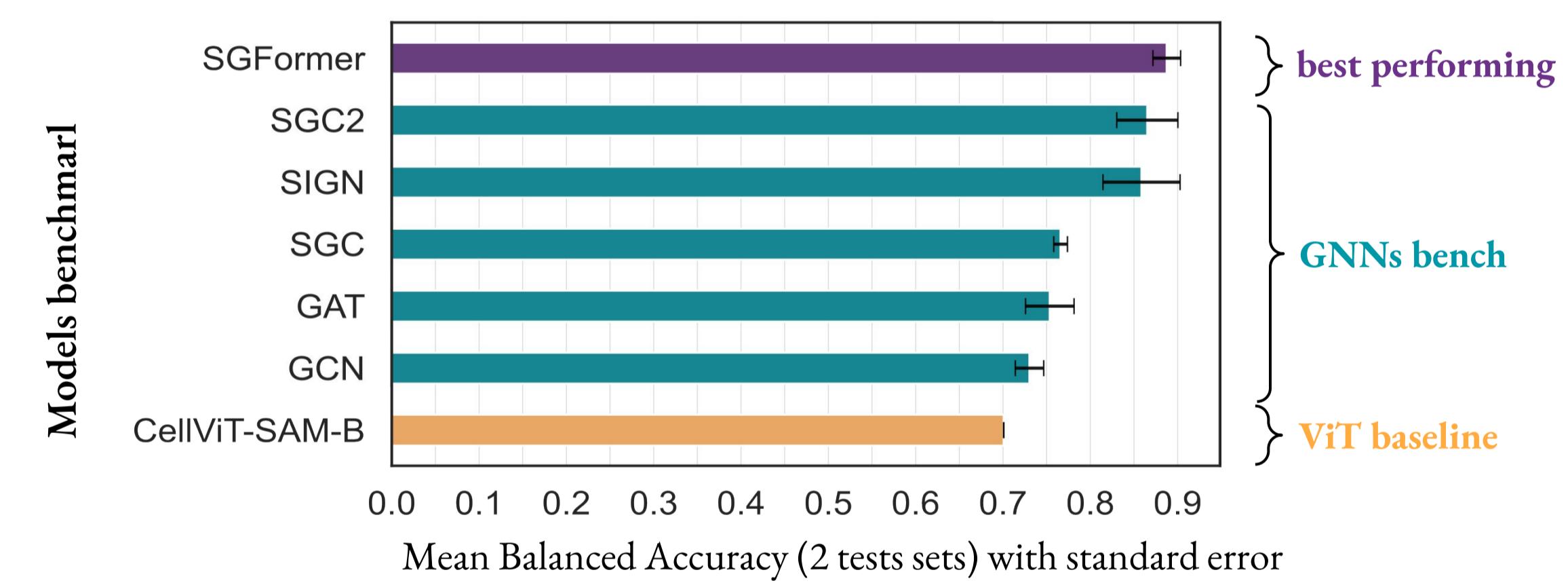
### METHOD

Use different Graph Neural Networks (GNNs), mainly SGFormer: very light Graph Transformer outperforming other GNNs for large graphs (100 000 > nodes):



Also housemade GNN in the works

### PRELIMINARY RESULTS



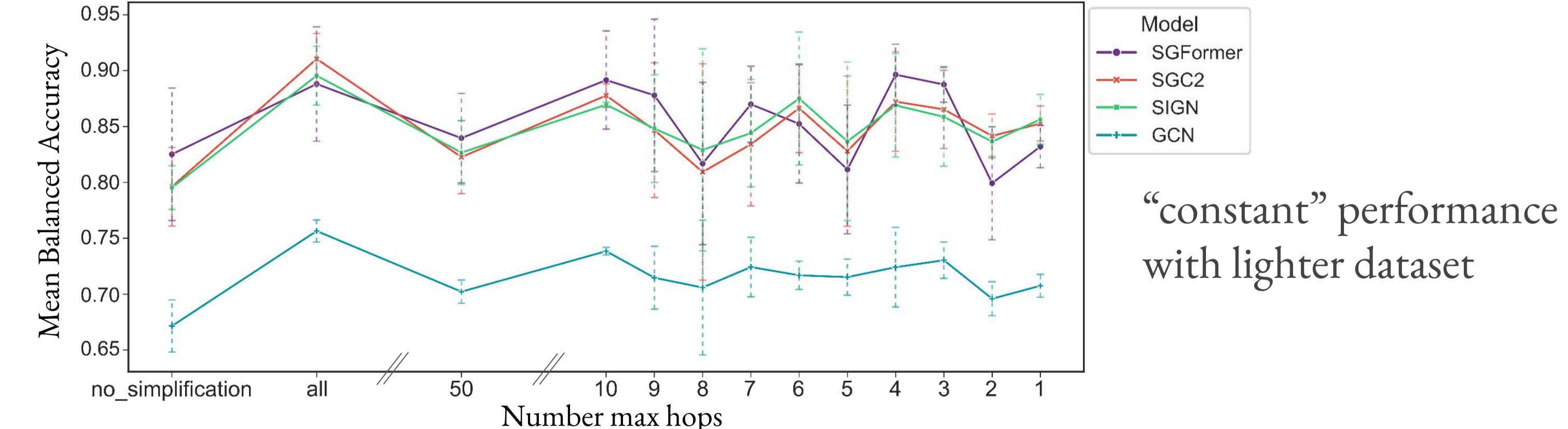
Comparison of mean balanced accuracy for different node features settings and splitting strategies on 2 test sets (with SGFormer model, italic: best training performance)

Node Features	Subgraphs Splits		Random Nodes Splits		
	with zscore-norm	3 hops max	no simplification	3 hops max	no simplification
Morphology only	No	0.811±0.001	0.873±0.004	0.960±0.000	0.956±0.002
Morphology only	Yes	0.776±0.015	0.895±0.070	0.970±0.001	0.970±0.001
Morphology+Texture	No	0.857±0.042	0.776±0.039	0.969±0.001	0.968±0.000
Morphology+Texture	Yes	0.897±0.013	0.853±0.031	0.981±0.000	0.972±0.001
Morphology+Texture+CellType	No	0.857±0.043	0.777±0.045	0.956±0.000	0.963±0.001
Morphology+Texture+CellType	Yes	0.888±0.016	0.825±0.059	0.968±0.001	0.979±0.001

The best performing models are in the case of **morphology + texture features with zscoring normalization** (ask me about the other interesting cases!)

Subgraphs splits and Random Node splits are not comparable trainings and evaluations. Random nodes is the standard method but introduce evaluation baises

Mean balanced accuracy for different models on 2 test sets GAT and SGC are not shown for readability



"constant" performance with lighter datasets

## NEXT STEPS

HP search with nested cross-validation

Working on a GNN with our new architecture

From 1 WSI / 1 graph to multiple graphs datasets