Supplementary Material for
Quantifying Explainers of Graph Neural Networks in Computational Pathology

Guillaume Jaume[1,2]*, Pushpak Pati[1,3,*], Behzad Bozorgtabar[2],
Antonio Foncubierta[1], Anna Maria Anniciello[4], Florinda Feroce[4], Tilman Rau[5],
Jean-Philippe Thiran[2], Maria Gabrani[1], Orcun Goksel[3,6]
[1]IBM Research Zurich, [2]EPFL Lausanne, [3]ETH Zurich,
[4]Fondazione Pascale, [5]University of Bern, [6]Uppsala University
{gja,pus}@zurich.ibm.com

1. Post-hoc explainers

In this section, we present the details of the considered graph explainability techniques (explainers) in this work: GRAPHLRP (Section 1.2), GRAPHGRAD-CAM (Section 1.3), GRAPHGRAD-CAM++ (Section 1.4), GNNEXPLAINER (Section 1.5), and RANDOM (Section 1.6).

1.1. Notation

We define an attributed undirected entity graph $G := (V, E, H)$ as a set of nodes V, edges E, and node attributes $H \in \mathbb{R}^{|V| \times d}$. d denotes the number of attributes per node, and $|.|$ denotes set cardinality. We denote an edge between nodes u and v as $e_{uv} \in E$. The graph topology is defined by a symmetric graph adjacency, $A \in \mathbb{R}^{|V| \times |V|}$, where $A_{uv} = 1$ if $e_{uv} \in E$. $H_{n,k}$ expresses the k-th attribute of the n-th node. The forward prediction of a cell-graph G_{CG} is denoted as, $y = \mathcal{M}(G_{\text{CG}})$, where $\mathcal{M}$ is a pre-trained GNN, and $y \in \mathbb{R}^{|\mathcal{T}|}$ are the output logits. Notation $y(t),\ t \in \mathcal{T}$ denotes the output logit of the t-th class. We refer to the logit of the predicted class as $y_{\max} = \max_{t \in \mathcal{T}} y(t)$, and the predicted class as $t_{\max} = \text{argmax}_{t \in \mathcal{T}}\, y(t)$.

1.2. Layerwise relevance propagation: GRAPHLRP

Layerwise Relevance Propagation (LRP) [1] is a feature attribution based post-hoc explainer. LRP explains an output logit by determining the individual contribution of each input element to the logit value. An output logit, defined as the output *relevance* for a given class, is layer-wise back-propagated until the input to compute the positive or negative impact of the input elements on the output logit. LRP, initially proposed for fully connected layers (LRP-FC), works as follows. Given a pre-trained fully connected layer $W \in \mathbb{R}^{z_1 \times z_2}$ between layer 1 and layer 2, where z_1 and z_2 are the number of neurons in layer 1 and layer 2, respectively, we compute the contributions of a neuron $i,\ i \in \{1, ..., z_1\}$ using the propagation rules in [6]. In this work, we are interested in identifying the input elements *positively* contributing to the prediction. To this end, we use the z^+ propagation rule that back-propagates the *positive* neuron contribution from layer 2 to layer 1 as:

$$R_i = \sum_j^{z_2} \frac{f_i |w_{ij}|}{\sum_k^{z_1} f_k |w_{kj}|} R_j \qquad \text{(LRP-FC)}$$

where $|w_{ij}|$ is the absolute value of the weight between i-th and j-th neuron in layer 1 and 2, respectively. f_i denotes the activation of the i-th neuron in layer l.

The extension from LRP-FC to LRP for graph isomorphism network (GIN) layers (GRAPHLRP) is achieved by following the observations in [8]. First, the *aggregate step* in GNN corresponds to projecting the graph's adjacency matrix on the node attribute space. For simplicity, assuming a 1-layer MLP as an update function, the GIN layer with *mean* aggregator can be re-written in its global form as:

$$H^{(l+1)} = \sigma\Big(W^{(l)}(I + \tilde{A})H^{(l)}\Big) \qquad (1)$$

where $\tilde{A}$ is the degree-normalized graph adjacency matrix, *i.e.* $\tilde{A}_{ij} = \frac{1}{|\mathcal{N}(i)|} A_{ij}$. σ is the ReLU activation function. Second, this representation allows us to treat the term $(I + \tilde{A})$ as a regular, fully connected layer. We can then apply the z^+ propagation rule with weights w_{ij} defined as:

$$w_{ij} = 1 \quad \text{if } i = j \qquad (2)$$

$$w_{ij} = \frac{1}{|\mathcal{N}(i)|} \quad \text{if } e_{ij} \in E \qquad (3)$$

$$w_{ij} = 0 \quad \text{otherwise} \qquad (4)$$

LRP outputs an importance score for each node i in the input graph.

*denotes equal contribution

1.3. Saliency-based: GRAPHGRAD-CAM

Grad-CAM [9] is a feature attribution post-hoc explainer that identifies salient regions of the input driving the neural network prediction. It assigns importance to each element of the input to produce Class Activation Map [11]. While originally developed for explaining CNNs operating on images, GRAD-CAM can be extended to GNNs operating on graphs [7].

GRAPHGRAD-CAM processes in two steps. First, it assigns an importance score to each channel of a graph convolutional layer. The importance of channel k in layer l is computed by looking at the gradient of the predicted output logit $y_{\max}$ w.r.t. the node attributes at layer l of the GNN. Formally it is expressed as:

$$w_k^{(l)} = \frac{1}{|V|} \sum_{n=1}^{|V|} \frac{\partial y_{\max}}{\partial H_{n,k}^{(l)}} \tag{5}$$

In the second step, a node-wise importance score is computed using the forward node feature activations $H^{(l)}$ as:

$$L(l, v) = \text{ReLU}\Big(\sum_{k}^{d^{(l)}} w_k^{(l)} H_{n,k}^{(l)} \Big) \quad \text{(GRAPHGRAD-CAM)}$$

where $L(l, v)$ denotes the importance of node $v \in V$ in layer l, and $d^{(l)}$ denotes the number of node attributes at layer l. Since we are only interested in the positive node contributions, *i.e.* nodes that positively influence the class prediction, we apply a ReLU activation to the node importances. Following prior work [7], we take the average node importance scores obtained over all the GNN layers $l \in \{1, ..., L\}$ to obtain smoother node importance scores.

1.4. Saliency-based: GRAPHGRAD-CAM++

GRAPHGRAD-CAM++ extends GRAD-CAM++ [2] to graph structured data. It improves the node importance localization by introducing node-wise contributions to channel importance scoring in Equation 5. Specifically, the modification is presented as,

$$w_k^{(l)} = \frac{1}{|V|} \sum_{n=1}^{|V|} \alpha_{n,k}^{(l)} \frac{\partial y_{max}}{\partial H_{n,k}^{(l)}} \tag{6}$$

where $\alpha_{n,k}^{(l)}$ are node-wise weights expressed for each attribute k at layer l. The derivation of a closed-form solution for $\alpha_{n,k}^{(l)}$ is analogous to the derivation in [2], where the size of graph, *i.e.* number of nodes, replaces the spatial dimensions of a channel as:

$$\alpha_{n,k}^{(l)} = \frac{\frac{\partial^2 y_{max}}{(\partial H_{n,k}^{(l)})^2}}{2\frac{\partial^2 y_{max}}{(\partial H_{n,k}^{(l)})^2} + \sum_{n=1}^{|V|} H_{n,k}^{(l)} \big(\frac{\partial^3 y_{max}}{(\partial H_{n,k}^{(l)})^3} \big)} \tag{7}$$

The subsequent node importance computation in GRAPHGRAD-CAM++ is same as GRAPHGRAD-CAM.

1.5. Graph pruning: GNNEXPLAINER

The GNNEXPLAINER [10, 5] is a graph pruning based post-hoc explainer for explaining GNNs. GNNEXPLAINER is model-agnostic, *i.e.* it can be used with any flavor of GNN. Intuitively, GNNEXPLAINER tries to find the minimum sub-graph $G_s \subset G$ such that the model prediction $y = \mathcal{M}(G)$ is retained. The inferred sub-graph G_s is then regarded as the *explanation* for G. This approach can be seen as a feature attribution method with *binarized* node importance scores, *i.e.* a node $v \in V$ has importance one if $v \in V_s$, and zero otherwise. Exhaustively searching G_s in the space created by nodes V and edges E is infeasible due to the combinatorial nature of the task. Instead, GNNEXPLAINER formulates the task as an optimization problem that learns a mask to activate or deactivate parts of the graph. The initial formulation by [10], developed for explaining node classification tasks, learns a mask over the edges, *i.e.* over the adjacency matrix. Instead, we follow the prior work in [5] to learn a mask over the nodes. Indeed, as we are concerned with classifying G, the optimal explanation G_s can be a disconnected graph. Furthermore, in cell graphs, the nodes representing biological entities are more intuitive and substantial for disease diagnosis than edges, that are heuristically-defined.

Formally, we seek to learn a mask M_V such that the induced masked sub-graph G_s, (1) is as small as possible, (2) outputs a binary node importance, and (3) provides the same prediction as the original graph. These constraints can be modeled by considering a loss function as:

$$\mathcal{L} = \mathcal{L}_{\text{KD}}(\hat{y}, y^{(m)}) + \alpha_{M_V} \sum_{i}^{|V|} \sigma(M_{V_i}^{(m)}) + \alpha_{\mathcal{H}} \mathcal{H}^e(\sigma(M_V^{(m)})) \tag{8}$$

where, m is the optimization step and σ is the sigmoid activation function. The first term is a knowledge-distillation loss $\mathcal{L}_{\text{KD}}$ between $\hat{y} = \mathcal{M}(G)$ and $y^{(m)} = \mathcal{M}(G_s)$ ensuring that $y^{(m)} \approx \hat{y}$. The second term aims to minimize the size of the mask M_V. The third term binarizes the mask by minimizing the element-wise entropy $\mathcal{H}^e$ of M_V. Following previous work [4], $\mathcal{L}_{\text{KD}}$ is built as a combination of distillation and cross-entropy loss,

$$\mathcal{L}_{\text{KD}} = \lambda \mathcal{L}_{\text{CE}} + (1 - \lambda)\mathcal{L}_{\text{dist}} \text{ where } \lambda = \frac{\mathcal{H}^e(y^{(m)})}{\mathcal{H}^e(\hat{y})} \tag{9}$$

where $\mathcal{L}_{\text{CE}}$ is the regular cross-entropy loss and $\mathcal{L}_{\text{dist}}$ is the distillation loss. When the element-wise entropy $\mathcal{H}^e(y^{(m)})$ increases, the term $\mathcal{L}_{\text{CE}}$ gets larger and reduces the probability of changing the prediction. Each term in Equation 8 is empirically weighed such that their contributions to $\mathcal{L}$

are comparable. We set $\alpha_{M_V} = 0.005$ and $\alpha_{\mathcal{H}} = 0.1$. We learn M_V using Adam optimizer with a learning rate of 0.01. $\mathcal{L}$ is optimized for 1000 steps with an early stopping mechanism, which triggers if the class prediction using G_s is changed. Therefore, G_s and G always predict the same class, *i.e.* $t_{\max}^{(m)} = \hat{t}_{\max}\ \forall m$.

1.6. Random selection: RANDOM

The RANDOM baseline is implemented using a *random* nuclei selection. The number of selected nuclei per RoI is given by the threshold value $k \in \mathcal{K}$.

2. BRACS dataset

In this paper, the BRACS dataset is used to analyze CG explainability for breast cancer subtyping. The pixel-level and entity-level statistics of the dataset are presented in Table 1. Training, validation, and test splits are created at the whole-slide level for conducting the experiments. The details of the class-wise distribution of images in each split are presented in Table 1.

3. Concepts and Attributes

In this paper, we focus on pathologically-understandable nuclear *concepts* $\mathcal{C}$ pertaining to nuclear morphology for breast cancer subtyping. To quantify each $c \in \mathcal{C}$, we use several measurable *attributes* $\mathcal{A}_c$. Table 2 presents the list of *concepts* and corresponding *attributes* used to perform the proposed quantitative analysis in this work. Also, Table 2 includes the class-wise expected criteria for each *concept*.

The *attributes* of the nuclei in a RoI are computed as presented in Table 2. It uses the RoI and corresponding nuclei segmentation map, denoted as I_{seg}. Area of a nucleus x, denoted as $A(x)$, is defined as the number of pixels belonging to x in I_{seg}. $P(x)$, the perimeter of x, is measured as the contour length of x in I_{seg}. $P_{\text{ConvHull}}(x)$, the convex hull perimeter of x, is defined as the contour length of convex hull induced by x in I_{seg}. The major and minor axis of x, noted as $a_{\text{major}}(x)$ and $a_{\text{minor}}(x)$, are the longest diameter of x and the longest line segment perpendicular to $a_{\text{major}}(x)$, respectively. The chromatin *attributes* are computed from the normalized gray level co-occurrence matrix (GLCM) [3], which captures the probability distribution of co-occurring gray values in x.

4. Quantitative assessment

In this section, we analyze two key components of the proposed quantitative metrics: the histogram construction and class separability scores for threshold set $\mathcal{K}$. Furthermore, we relate the analysis to the class-wise expected criteria for each *concept* presented in Table 2.

4.1. Histogram analysis

Histogram construction is a key component in the proposed quantitative metrics. Figure 1 presents per-class histograms for each explainer and the best *attribute* per *concept*. We set the importance threshold to $k = 25$, *i.e.* for each RoI, we select 25 nuclei with the highest node importance. The best *attribute* for a *concept* is the one with the highest average pair-wise class separability.

The row-wise observation exhibits that GNNEXPLAINER and GRAPHLRP provide, respectively, the maximum and the minimum pair-wise class separability. The histograms for a *concept* and for an explainer can be analyzed to assess the agreement between the selected important nuclei *concept*, and the expected *concept* behavior as presented in Table 2, for all the classes. For instance, nuclear *area* is expected to be higher for malignant RoIs than benign ones. The *area* histograms for GNNEXPLAINER, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ indicate that the important nuclei set in malignant RoIs includes nuclei with higher area compared to benign RoIs. Similarly, the important nuclei in malignant RoIs are expected to be vesicular, *i.e.* high texture entropy, compared to light euchromatic, *i.e.* moderate texture entropy, in benign RoIs. The *chromaticity* histograms for GNNEXPLAINER, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ display this behavior. Additionally, the histogram analysis can reveal the important *concepts* and important *attributes*. For instance, nuclear *density* proves to be the least important *concept* for differentiating the classes.

4.2. Separability score for threshold set $\mathcal{K}$

Multiple importance thresholds $\mathcal{K}$ are required to address the varying notion of important nuclei across different cell graphs and different explainers. Figure 2 presents the behavior of pair-wise class separability for using various $k \in \mathcal{K} = \{5, 10, ..., 50\}$. For simplicity, we present the behavior for the best *attribute* per *concept*. In general, the pair-wise class separability is observed to decrease with decreasing k. Intuitively, decreasing k results in including more unimportant nuclei into the evaluation, thereby gradually decreasing the class separability.

The degree of agreement between the difference in the expected behavior per *concept* and the pair-wise class separability in Figure 2, for all pair-wise classifications and various $k \in \mathcal{K}$ can be used to assess the explainer's quality. For instance, according to Table 2, the difference in the expected nuclear *size* can be considered as benign–atypical $<$ benign–malignant, and atypical–malignant $<$ benign–malignant. GNNEXPLAINER, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ display these behaviors $\forall k \in \mathcal{K}$. GNNEXPLAINER provides the highest class separability in each pair-wise classification, thus proving to be the best ex-

	Metric	Benign	Atypical	Malignant	Total
Image	Number of images	1741	1351	1299	4391
	Number of pixels (in million)	3.9±3.54	1.62±1.48	6.35±5.2	3.9±4.3
	Max/Min pixel ratio	180.1	75.3	128.6	235.6
CG	Number of nodes	1331±1134	635±510	2521±1934	1468±1642
	Number of edges	4674±4131	2309±2110	8591±7646	5102±6089
	Max/Min node ratio	312.5	416.7	312.5	434.8
Image split	Train	1231	1008	928	3163
	Validation	261	162	179	602
	Test	249	185	192	626

Table 1. Statistics of BRACS dataset.

Concept ($\mathcal{C}$)	Attribute ($\mathcal{A}$)	Computation	Benign	Atypical	Malignant
Size	Area	$A(x)$	Small	Small-Medium	Medium-Large
Shape	Perimeter	$P(x)$	Smooth	Mild irregular	Irregular
	Roughness	$\frac{P_{\text{ConvHull}}(x)}{P(x)}$			
	Eccentricity	$\frac{a_{\text{minor}}(x)}{a_{\text{major}}(x)}$			
	Circularity	$\frac{4\pi A(x))}{P(x)^2}$			
Shape variation	Shape factor	$\frac{4\pi A(x)}{P^2_{\text{ConvHull}}}$	Monomorphic	Monomorphic	Pleomorphic
Spacing	Mean spacing	mean$(d_y \vert y \in \text{kNN(x)})$	Evenly crowded	Evenly spaced	Variable
	Std spacing	std$(d_y \vert y \in \text{kNN(x)})$			
Chromatin	GLCM dissimilarity	$\sum_i \sum_j \vert i-j \vert p(i,j)$	Light euchromatic	Hyperchromatic	Vesicular
	GLCM contrast	$\sum_i \sum_j (i-j)^2 p(i,j)$			
	GLCM homogenity	$\sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2}$			
	GLCM ASM	$\sum_i \sum_j p(i,j)^2$			
	GLCM entropy	$-\sum_i \sum_j p(i,j) \log(p(i,j))$			
	GLCM variance	$\sum_i \sum_j (i-\mu_i)^2 p(i,j)$ with $\mu_i = \sum_i \sum_j i p(i,j)$			

Table 2. Pathologically-understandable nuclear *concepts*, corresponding measurable *attributes*, and computations are shown in Columns 1, 2, 3, respectively. The expected *concept* behavior for three breast cancer subtypes is shown in Columns 4, 5, 6, respectively.

plainer pertaining to *size concept*. Detailed inspection of Figure 2 shows that all the differences in the expected behavior, per *concept* for all pair-wise classifications, is inline with the *concept*-wise expected behavior in Table 2, $\forall c \in \mathcal{C}$ and $\forall k \in \mathcal{K}$. Overall, GNNEXPLAINER is seen to be the best explainer as it agrees to the majority of the expected differences $\forall c \in \mathcal{C}$ for all pair-wise classifications, while providing high-class separability. Furthermore, *size* proves to be the most important *concept* that provides the maximum class separability across all pair-wise classifications.

5. Qualitative assessment

Figure 3 and Figure 4 present CG explanations produced by GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++ and GRAPHLRP for RoIs across benign, atypical and malignant breast tumors. It can be observed that GNNEXPLAINER learns to binarize the explanations, thereby producing the most compact explanations by retaining the most important nuclei set of nuclei with high importance. However, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ produce explanations with more distributed nuclei importance than GNNEXPLAINER.

GRAPHLRP produces the largest explanations by retaining most of the nuclei in the CGs.

References

[1] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7), 2015. 1

[2] A. Chattopadhay, A. Sarkar, P. Howlader, and V.N. Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*, volume 2018-Janua, pages 839–847, 2018. 2

[3] R.M. Haralick, K. Shanmugam, and I. Dinstein. Texural features for image classification. *IEEE transaction on systems, man and cybernatics*, 3(6):610–621, 1973. 3

[4] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems*, 2015. 2

[5] G. Jaume, P. Pati, A.F. Rodriguez, F. Florinda, G. Scognamiglio, A.M. Anniciello, J.P. Thiran, O. Goksel, and M. Gabrani. Towards Explainable Graph Representations in Digital Pathology. In *International Conference on Machine Learning Workshops*, 2020. 2

[6] G. Montavon, S. Bach, A. Binder, W. Samek, and K.R. Muller. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition. *Pattern Recognition*, 65:211–222, 2015. 1

[7] P.E. Pope, S. Kolouri, M. Rostami, C.E. Martin, and H. Hoffmann. Explainability methods for graph convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10764–10773, 2019. 2

[8] R. Schwarzenberg, M. Huebner, D. Harbecke, C. Alt, and L. Hennig. Layerwise relevance visualization in convolutional text graph classifiers. *EMNLP Workshop*, pages 58–62, 2019. 1

[9] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, and D. Batra. Grad-CAM : Visual Explanations from Deep Networks. In *International Conference on Computer Vision*, pages 618–626, 2017. 2

[10] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. GNNExplainer: Generating Explanations for Graph Neural Networks. In *Advances in Neural Information Processing Systems*, 2019. 2

[11] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. 2

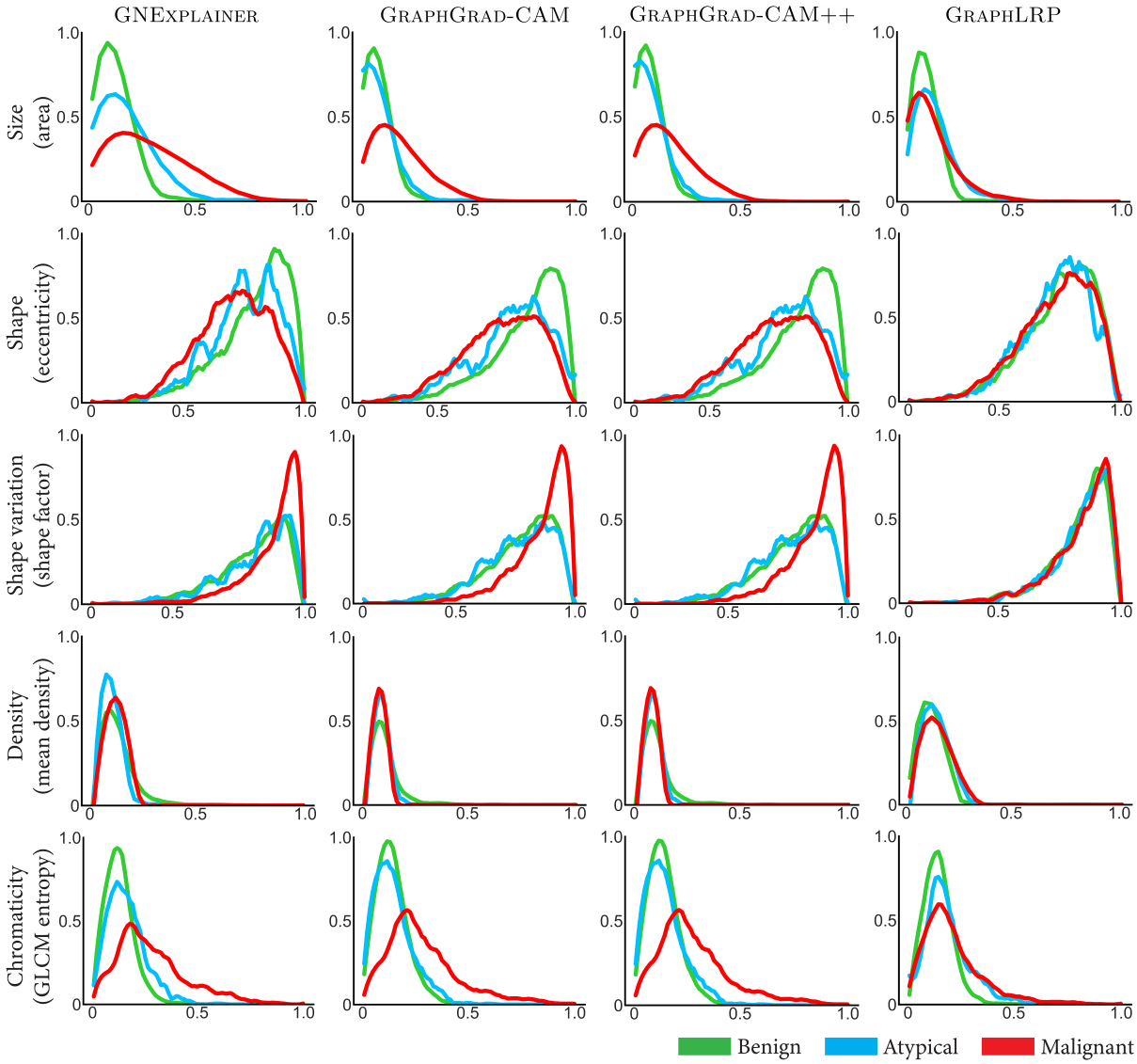

Figure 1. Per-class histograms for different *concepts* across different graph explainers. For simplicity, histograms are presented for the best *attribute* per *concept* at fixed importance threshold $k = 25$.

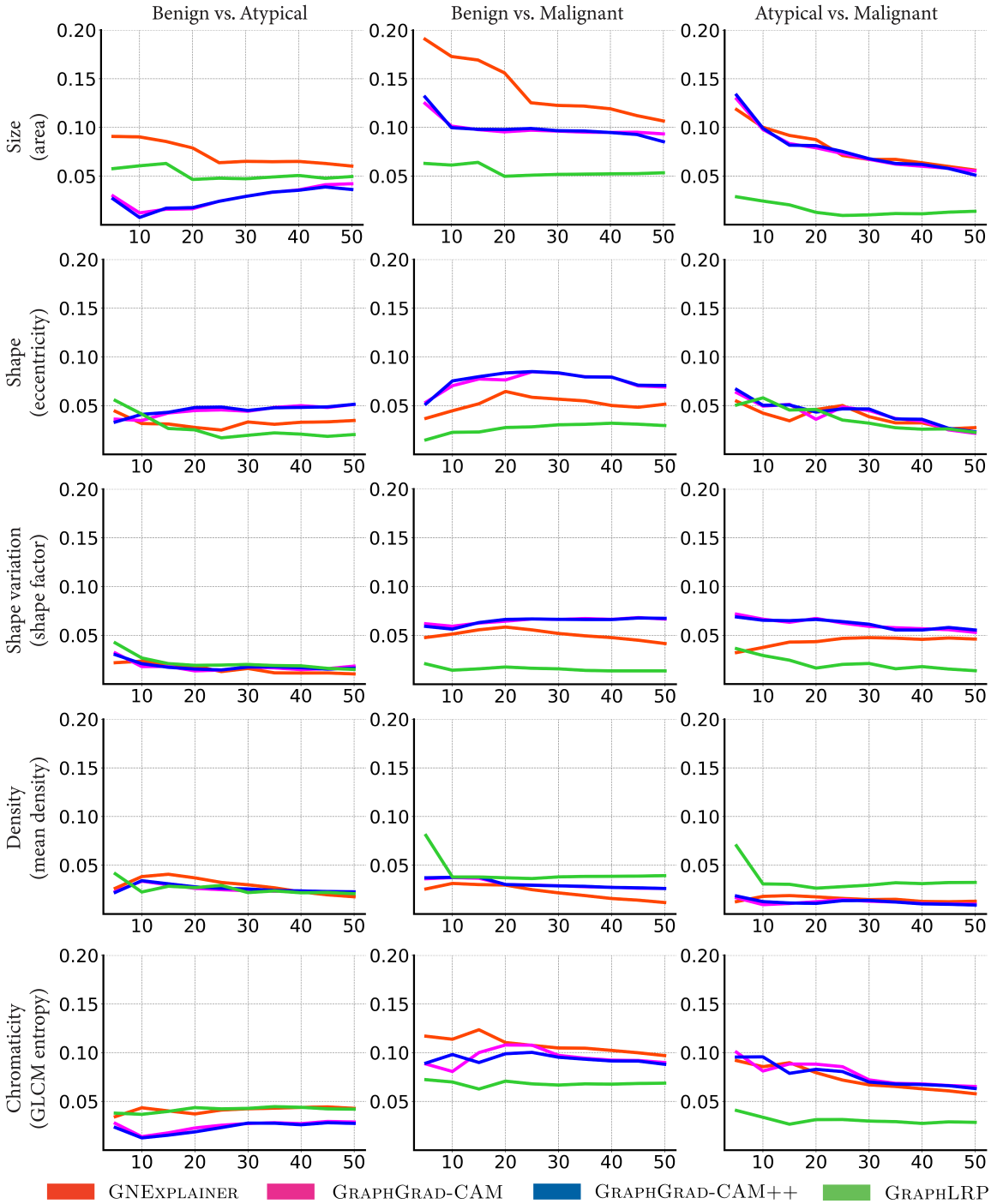

Figure 2. Visualizing the variation of pair-wise class separability score (Y-axis) w.r.t. various nuclei importance thresholds in $\mathcal{K}$ (X-axis). The analysis is provided for different graph explainers, and for the best *attribute* per *concept*.

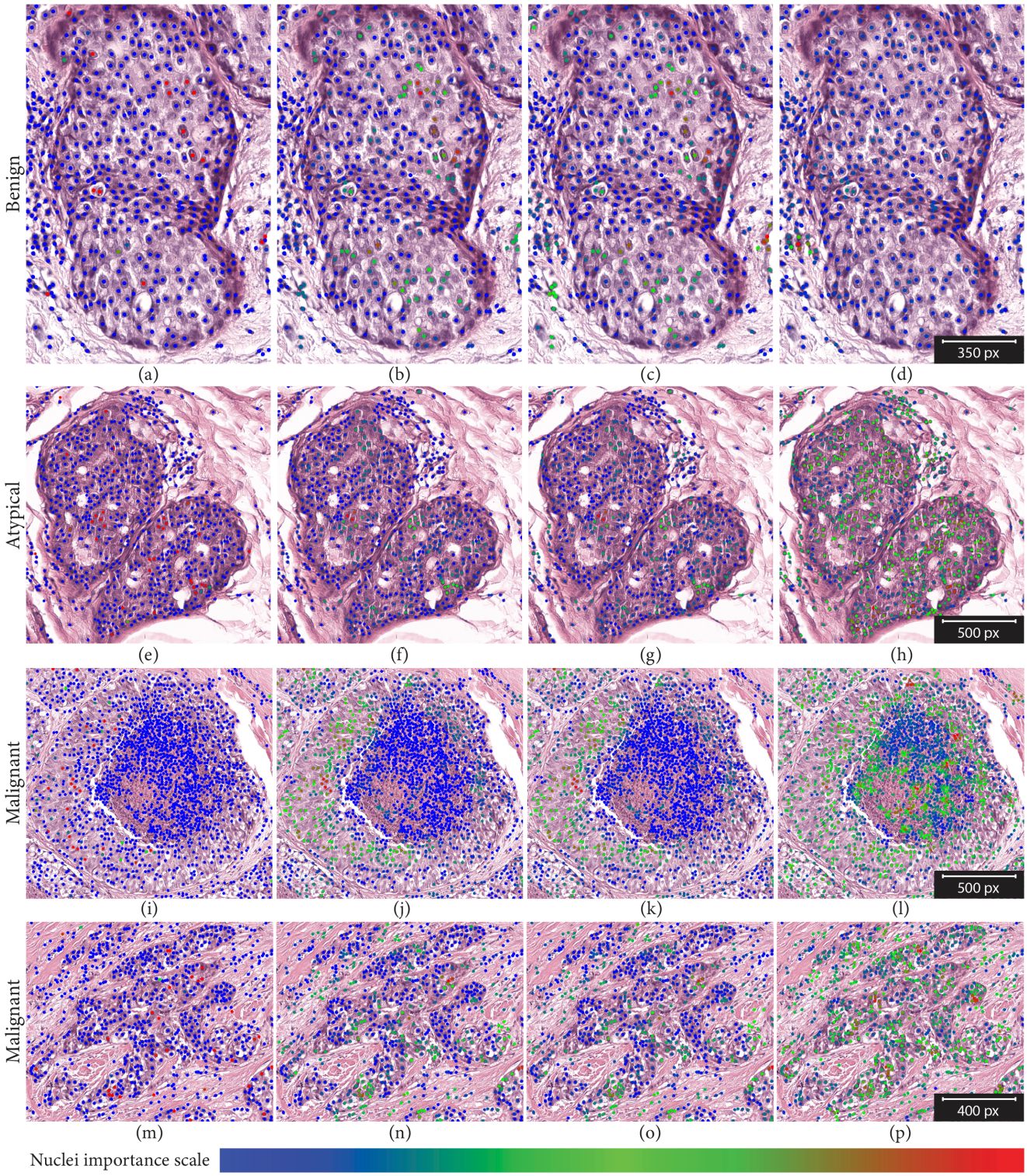

Figure 3. Qualitative results. The rows represent breast cancer subtypes, and columns represent graph explainers, *i.e.* GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++, and GRAPHLRP. Nuclei level importance ranges from blue (the least important) to red (the highest important).

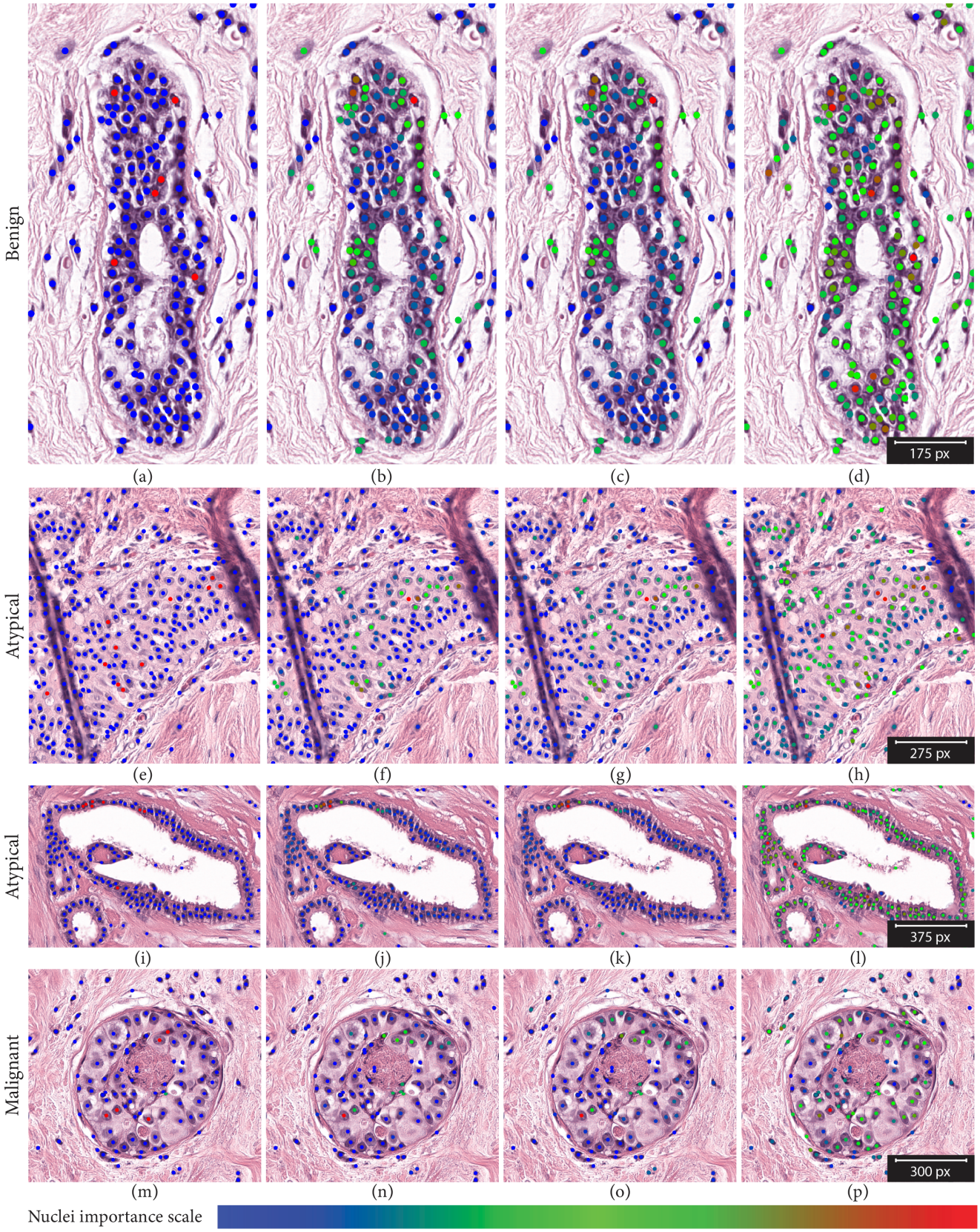

Figure 4. Qualitative results. The rows represent breast cancer subtypes, and columns represent graph explainers, *i.e.* GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++, and GRAPHLRP. Nuclei level importance ranges from blue (the least important) to red (the highest important).