scientific reports

Check for updates

OPEN

# Distilling knowledge from graph neural networks trained on cell graphs to non-neural student models

Vasundhara Acharya[1]✉, Bülent Yener[2] & Gillian Beamer[3]

**The development and refinement of artificial intelligence (AI) and machine learning algorithms have been an area of intense research in radiology and pathology, particularly for automated or computer-aided diagnosis. Whole Slide Imaging (WSI) has emerged as a promising tool for developing and utilizing such algorithms in diagnostic and experimental pathology. However, patch-wise analysis of WSIs often falls short of capturing the intricate cell-level interactions within local microenvironment. A robust alternative to address this limitation involves leveraging cell graph representations, thereby enabling a more detailed analysis of local cell interactions. These cell graphs encapsulate the local spatial arrangement of cells in histopathology images, a factor proven to have significant prognostic value. Graph Neural Networks (GNNs) can effectively utilize these spatial feature representations and other features, demonstrating promising performance across classification tasks of varying complexities. It is also feasible to distill the knowledge acquired by deep neural networks to smaller student models through knowledge distillation (KD), achieving goals such as model compression and performance enhancement. Traditional approaches for constructing cell graphs generally rely on edge thresholds defined by sparsity/density or the assumption that nearby cells interact. However, such methods may fail to capture biologically meaningful interactions. Additionally, existing works in knowledge distillation primarily focus on distilling knowledge between neural networks. We designed cell graphs with biologically informed edge thresholds or criteria to address these limitations, moving beyond density/sparsity-based definitions. Furthermore, we demonstrated that student models do not need to be neural networks. Even non-neural models can learn from a neural network teacher. We evaluated our approach across varying dataset complexities, including the presence or absence of distribution shifts, varying degrees of imbalance, and different levels of graph complexity for training GNNs. We also investigated whether softened probabilities obtained from calibrated logits offered better guidance than raw logits. Our experiments revealed that the teacher's guidance was effective when distribution shifts existed in the data. The teacher model demonstrated decent performance due to its higher complexity and ability to use cell graph structures and features. Its logits provided rich information and regularization to students, mitigating the risk of overfitting the training distribution. We also examined the differences in feature importance between student models trained with the teacher's logits and their counterparts trained on hard labels. In particular, the student model demonstrated a stronger emphasis on morphological features in the Tuberculosis (TB) dataset than the models trained with hard labels. This emphasis aligns closely with the features that pathologists typically prioritize for diagnostic purposes. Future work could explore designing alternative teacher models, evaluating the proposed approach on larger datasets, and investigating causal knowledge distillation as a potential extension.**

**Keywords** Whole slide imaging, Graph neural networks, Cell graphs, Knowledge distillation, Non-neural models, Tuberculosis

Cell graphs have emerged as a powerful tool for capturing the spatial and functional relationships within tissues. They encapsulate cellular and tissue-level architecture by representing cells as nodes and their interactions as

[1]Rensselaer Polytechnic Institute, Troy, USA. [2]Professor, Rensselaer Polytechnic Institute, Troy, USA. [3]Adjunct Associate Professor, Texas Biomedical Research Institute, San Antonio, TX, USA. ✉email: acharv2@rpi.edu

edges. They are particularly valuable for bridging the gap between molecular details and their collective impact on larger biological processes, such as wound healing, tumor progression, and immune response[1]. The cell-graph technique seeks to uncover the structure-function relationship by modeling the structural organization of tissue using graph theory. For instance, in the context of breast cancer, cancer cells often cluster together to form dense regions of abnormal tissue. This clustering reflects the biological processes underlying tumor growth[2], such as rapid cell division, altered adhesion properties, and disrupted tissue architecture. By analyzing the spatial distribution and interactions of these clustered cells, the cell-graph approach can provide insights into the functional state of the tissue, such as tumor aggressiveness. This study focuses on three primary cell graph-based datasets: Tuberculosis (TB), Placenta, and Breast Cancer Classification. TB is a highly contagious disease and a leading cause of ill health and mortality worldwide. According to the World Health Organization's report on TB[3], an estimated 1.25 million people succumbed to the disease in 2023. Pulmonary TB, primarily caused by an infectious bacterium, predominantly impacts the lungs through airborne transmission[4]. Granulomas in lung tissue are characteristic of both human and experimental pulmonary tuberculosis[5,6]. Identifying acid-fast bacilli (AFB) in stained samples is essential for diagnosing tuberculosis[7].

Whole-slide imaging makes it easier to digitally examine these stained samples, allowing for high-resolution, in-depth tissue investigation. They preserve fine-grained cellular morphology and local tissue architecture that is often lost through downsampling. Traditional WSI analysis pipelines resort to patch-based processing or downsampling, fragmenting tissue structure and sacrificing essential contextual information[8]. In our approach, we construct cell graphs from whole slide images that integrate local morphological features with spatial context. A deep GNN is then applied to these graphs to learn complex cell interactions, translating the rich WSI content into structured, relational representations. The edge threshold for intercellular communication is crucial in constructing biologically meaningful cell graphs. Incorporating pathologist insights can help refine this threshold, ensuring the graph representation aligns with the underlying cellular interactions. We determined edge thresholds based on the biological rationale for the cell graphs we constructed and validated them through consultations with our domain expert. For the TB dataset cell graphs, nodes represent either acid-fast bacilli (AFB) or the nucleus of activated macrophages, and edge thresholds are based on the length of cords of the *M.tb* infected cells after 72 hours of infection[9] and the fact that macrophages extend pseudopods to sense their environment[10]. The Placenta dataset represents diverse histological structures essential to placental biology, including various types of trophoblastic villi (TVilli, MIVilli, SVilli, AVilli), Sprouts, Chorion, Maternal cells, Fibrin, and Avascular regions. These structures capture key functional and structural aspects of the placenta. Cell graphs from this dataset reveal how these structures collectively contribute to placental function. Finally, cell graphs from the breast cancer dataset show the spatial arrangement and interactions between tumor cells, lymphocytes, and stromal cells. Edges in these cell graphs were constructed based on factors such as immune surveillance by lymphocytes and the clustering behavior of tumor cells facilitated by adhesion molecules[11]. They captured important patterns, such as tumor-immune interactions and interactions with stromal cells, essential for understanding disease progression and prognosis.

The cell-graph technique leverages image processing, feature extraction, and machine learning algorithms to establish a quantitative relationship between structure and function[1]. Our approach extends this by employing a GNN trained on these cell graphs to learn and model this relationship effectively. Within our proposed graph model, which we term as Cell Graph Jumping Knowledge Neural Network (CG-JKNN), we incorporate the concept of 'jumping knowledge'[12] from GraphSAGE layers. This approach aggregates information from multiple network layers rather than relying solely on the final layer. We enhance the jumping knowledge with GATv2's attention mechanism to refine this process further. This allows the model to focus on the most informative nodes dynamically.

An important question is whether the knowledge learned by complex deep learning models, such as GNNs in our work, can be effectively distilled into simpler, non-neural network-based models. The answer lies in knowledge distillation (KD), a process where the knowledge from a teacher model (in this case, a GNN) is distilled into student models, typically less complex. Knowledge distillation on graphs brings the advantages of KD into graph learning. This approach primarily serves two objectives: model compression and performance improvement. Model compression focuses on creating a smaller student model than the teacher model. After distillation, the student model achieves a performance comparable to that of the teacher while requiring fewer parameters. Performance improvement focuses on transferring knowledge from the teacher to the student model, aiming to enhance the student's performance beyond that of a model trained without knowledge distillation[13]. The student model may be smaller, similar, or architecturally different from the teacher. The other main goals of KD are knowledge adaptation and knowledge expansion[14]. Knowledge adaptation focuses on helping student networks perform well on new, unseen target domains by using knowledge from teacher networks trained on similar source domains. Knowledge expansion aims to create student networks that are more capable and perform better than the teacher networks. In our work, we focus on model compression and performance improvement. Existing approaches to knowledge distillation mainly focus on neural network-based student models[15–17] using their iterative learning capabilities to align with the teacher's outputs. However, this work demonstrates that knowledge can be distilled to non-neural network-based models, such as tree-based ensemble models. The knowledge that can be distilled can be categorized into various forms, including response-based, intermediate, relation-based, and mutual information-based representations[14]. In this work, we focus on response-based knowledge distillation, using the logits generated by a deep GNN as targets to train tree-based ensemble regressor models. These student models are significantly less complex than the teacher. Our primary objective is to evaluate whether the teacher's guidance through logits provides better insights into the student models than traditional hard labels. We will use the term "Guidance" throughout the paper, which refers to the teacher model's ability to provide detailed class distinctions and enhance the student model's performance and generalization through its logits. Literature suggests that students trained on logits are better equipped to

mimic the behavior of the teacher model[18]. This approach enhances the student's performance and enables it to be a partial proxy for interpreting the teacher's decision-making process. In one of our ablation studies, we analyze the differences in feature importance between the student trained on logits and its counterpart trained on hard labels to identify any notable distinctions. To measure the efficacy of this distillation process, we employ a distillation quality metric that balances model complexity and performance. Furthermore, we extend our analysis to explore whether calibration (aligning the probabilities derived from logits with the true likelihood of events) improves the guidance provided by logits. Additionally, we evaluate the efficacy of our approach under varying dataset complexities, including the presence or absence of distribution shifts, imbalanced data, different feature sets, and different levels of training graph complexity. To broaden the applicability of our method, we also test it on datasets beyond cell graphs.

In this study, we addressed key questions to learn the efficacy of knowledge distillation in our proposed framework. Specifically, we sought to answer the following:

- Do all student models benefit from knowledge distilled from the teacher GNN trained on cell graphs with local cell graph features and/or morphological features under varying dataset complexities such as the presence of distribution shifts?
- Do the features selected by models trained on hard labels differ from those chosen by the students, and can these differences provide insights into the teacher's guidance?
- Can a student model achieve better performance when trained using the combined guidance of the teacher model and the best-performing student, compared to being taught solely by the teacher model?
- Can calibration of teacher logits provide better guidance to student models?

The major contributions of this work can be summarized as follows:

- Inspired by Fukui et al.[19], we proposed a knowledge distillation framework that uses the logits from a GNN model with jumping knowledge, which acts as the teacher, to train non-neural network models as student models. To our knowledge, this is the first work exploring a teacher trained on cell graphs to guide non-neural network-based student models.
- We proposed a method to approximate the number of parameters/complexity of student models using the asymptotic equivalence between the Akaike Information Criterion (AIC) and leave-one-out cross-validation.
- We evaluated the efficacy of knowledge distillation under diverse dataset conditions, including varying degrees of imbalance, distribution shifts, and varying graph complexities. We also tested our approach across various feature sets, including combinations of cell graph features and morphological features, individual feature sets (only cell graph features or morphological features), and non-cell graph features.
- We explored the impact of post-calibrating logits to enhance the guidance provided by teacher models to student models. We proposed a modified distillation quality metric that effectively measures the quality of knowledge distilled, even in scenarios where the student model outperforms the teacher.
- We conducted ablation studies to determine whether the best-performing student model, in combination with the teacher model, could improve guidance. Additionally, we analyzed how feature importance varied when guided by the teacher and explored the biological relevance of these features.

Section "Related works" discusses prior research in the domain. Section "Methods" describes this study's proposed methodology and framework. Section "Results" presents the experimental results and evaluates the performance of our approach. Section "Discussion and major takeaways" analyzes the implications of our findings and summarizes the key takeaways of this study. Section "Limitations of our work" outlines the limitations of our approach. Section "Conclusion and future work" summarizes the contributions and identifies areas for future work.

## Related works

### Cell graphs and GNNs trained on cell graphs: applications in disease prediction and classification

Graph construction for modeling cellular interactions often assumes that neighboring cells are more likely to interact. To capture these interactions, methods such as Delaunay triangulation[1,20,21] and K-nearest-neighbor (KNN)[22–25] are widely employed. The Waxman model[26] is another approach that uses an exponential decay function of Euclidean distance to define edges probabilistically. Numerous studies have utilized cell graphs to gain insights into the organization and behavior of cells within tissues. The pioneering work on cell graphs highlighted that the most effective cell-graph construction methods emerge from combining physics-driven and data-driven paradigms[1]. The study presented in[27] used a computational method using cell-graph evolution to model glioma malignancy. It linked graph phases to cancer severity through connectivity analysis of cell graphs constructed from tissue photomicrographs. The authors in[28] presented a computational method to model glioma malignancy using cell-graph topology from tissue images. Cell-graph edges were generated using the Waxman model. By analyzing graph metrics of cancerous cell clusters, the method achieved 85% accuracy at the cellular level and 100% accuracy at the tissue level. An augmented cell-graph (ACG) method for diagnosing malignant glioma from low-magnification tissue images was introduced in[29]. It represented cell clusters as nodes and their relationships as weighted edges. Tested on 646 brain biopsy samples, the approach achieved 97.53% sensitivity and specificities of 93.33% (inflamed) and 98.15% (healthy) at the tissue level. Gunduz-Demir[30] introduced an object-graph-based approach for gland segmentation by leveraging the organizational properties of primitive objects. It achieved high segmentation accuracy when applied to colon tissue images and demonstrated robustness to artifacts and tissue variances. The authors in[31] introduced a Cell Graph Transformer

(CGT) for nuclei classification in histopathology images. A topology-aware pretraining method using a graph convolutional network (GCN) was proposed to learn a feature extractor to address challenges with noisy self-attention scores in complex cell graphs. The study in[32] presented sigGCN, a multimodal deep learning model combining a graph convolutional network (GCN) and neural network to integrate gene interaction networks for cell classification. The method outperformed existing traditional approaches in both within-dataset and cross-dataset classifications. Graph neural network-based approach that leveraged cell graphs from multiplexed immunohistochemistry (mIHC) images to predict patient survival and digitally stage gastric cancer was proposed in[33]. Edges in the cell graph were established based on the Euclidean distance between cell pairs, connecting cells separated by less than 20 $\mu$m. It outperformed traditional staging systems, achieving high AUC scores (0.960 for binary and 0.771-0.904 for ternary classification). A novel cell-graph convolutional neural network for colorectal cancer (CRC) grading that models large histology images as graphs was proposed in[23]. It incorporated both nuclear appearance and spatial information. An edge was placed between two nuclei if they were at a fixed distance from each other. By introducing Adaptive GraphSage for multi-scale feature fusion and a sampling technique to address graph redundancy, CGC-Net effectively captured tissue micro-environment structures. A hierarchical Transformer Graph Neural Network, combining GNN and Transformer architectures, was introduced in[24]. The main aim was to achieve colorectal adenocarcinoma cancer (CRA) grading using the cell graph that was constructed using the KNN approach. It used a Masked Nuclei Patch (MNP) strategy to train a ResNet-50 to extract representative nuclei features. The transformer module captured long-distance dependencies, achieving state-of-the-art results on CRA grading tasks. The authors in[34] proposed Feature-Driven Local Cell Graphs (FeDeG) for constructing cell graphs by integrating spatial proximity and nuclear attributes like shape, size, and texture. Graph-derived metrics extracted from FeDeGs were used with a linear discriminant classifier, achieving an AUC of 0.68. A Hierarchical Cell-to-Tissue (HACT) graph representation utilizing the cell graphs was proposed in[35]. The tissue structure and functionality were modeled using a novel hierarchical graph neural network (HACT-Net). Using the Breast Carcinoma Subtyping (BRACS) dataset, HACT-Net outperformed state-of-the-art methods and individual pathologists.

## Knowledge distillation in graphs

With the demand for efficient models, KD is an ever-developing field. Among the various types of information that can be distilled, including logits, embeddings, and graph structures, we specifically use logits as the training labels for the student models. Many works have focused on transferring logits as a form of knowledge in knowledge distillation. The authors in[36] systematically compared different knowledge sources–features, logits, and gradients in knowledge distillation by approximating the KL-divergence criterion. They analyzed their effectiveness in model compression and incremental learning and found that logits were generally more efficient. Recently, a refined knowledge distillation method that employed labeling information to refine teacher logit dynamically and to eliminate misleading information from the teacher was introduced in[37]. Distilling graph structure information involves transferring knowledge about the connectivity and relationships between nodes and edges[38], which is crucial for modeling graph data. Additionally, some works distill learned node embeddings from the intermediate layers of teacher models to guide the student model's learning.

In the context of knowledge distillation, various setups exist to transfer knowledge. There are teacher-free networks where the student model learns independently without a teacher. In teacher-to-student networks, the knowledge transfer can involve one or multiple teachers guiding the students. Additionally, distillation can be categorized as offline or online. Online distillation refers to a scheme where the teacher and student models are trained simultaneously in an end-to-end manner. In contrast, offline distillation involves a pre-trained teacher model that facilitates the student's training without undergoing further updates. In our study, we utilize a teacher-to-student setup with two configurations: a single teacher guiding the student and a combination of the teacher and the best-performing student acting as teachers. Additionally, our approach falls under the category of offline distillation, as the teacher models are pre-trained and remain unchanged during the training of the student models.

Numerous works have been conducted to highlight the use of knowledge distillation in graphs. In[39], the authors proposed a method for compressing a $k$-layered graph convolution network (GCN) by repeating a single GCN layer $k$ times and distilling both the logits and final node embeddings. The authors in[40] used two heterogeneous teacher models to distill their embeddings via a topological attribution map and logits. In[41], the authors trained a teacher on offline graph snapshots with a self-attention mechanism to distill to a smaller, more efficient student model making predictions on online graph snapshots. A neighbor distillation method to distill local structure knowledge and to use peer node information to learn the local structure was proposed in[42]. The approach in[43] used logit distillation and auxiliary representation distillation methods such as Locality Structure Preserving distillation (LSP)[44]. In[45], the authors used adversarial training for KD by applying a discriminator to the embeddings and logits of the student and teacher models. The authors in[46] proposed a method for fair distillation where a student model learned both the distilled logits and a proxy for bias from the teacher, which was removed during testing with the rationale that it contained most of the information on bias and its exclusion would result in fair predictions. An interesting logits-based KD method termed Decoupled Graph Knowledge Distillation (DGKD) was proposed in[47]. It reformulated the distillation loss into the components of target class (TCGD) and non-target class (NCGD). By decoupling the fixed weight between these losses and addressing their negative correlation, DGKD dynamically adjusted the weights for different data samples. This led to improved prediction accuracy for student MLP. The authors in[48] proposed Knowledge Distillation for Graph Augmentation (KDGA) that mitigated the adverse effects of distribution shifts caused by graph augmentation. KDGA transferred knowledge from a GNN teacher trained on augmented graphs to a partially parameter-shared student tested on the original graph. This helped to improve performance across various GNN architectures and augmentation methods. In[49], they transferred knowledge from two specialized teacher models, one focused on

features and the other on structure, using a teacher-student distillation framework. The feature-level teacher guided the student on completing and leveraging node features, while the structure-level teacher focused on graph topology. However, these works primarily focused on distilling knowledge from a GNN to another GNN or other neural networks. In[19], the authors proposed a distillation method that utilized information extracted from neural networks to train non-neural network models, such as support vector machines, random forests, and gradient-boosting decision trees. Their study was limited to a single image-based dataset and did not provide a detailed analysis of why specific student models failed to achieve the desired performance when trained with logits obtained from the teacher CNN. Moreover, they evaluated their approach using only two out of ten available classes for simplicity, which does not adequately demonstrate the efficacy of KD in a multiclass setting.

### Problem statement

Currently used methods for building cell graphs typically use a single-edge threshold to represent every interaction between cells. These thresholds are often chosen based on factors such as achieving denser graphs. However, this approach overlooks the biological diversity of interactions, as different cell types exhibit distinct interaction patterns that a uniform edge threshold cannot adequately capture. A more biologically informed methodology for defining these thresholds is necessary to better reflect the underlying cellular relationships. In the context of knowledge distillation from GNNs, most existing works focus on transferring knowledge from GNNs to other neural networks. However, student models need not be limited to neural networks. They can include non-neural models. Furthermore, evaluating the efficacy of knowledge distillation in our specific setup requires a broader understanding of its behavior under varying dataset complexities, including scenarios with distribution shifts, multiple classes, and other challenges.

## Methods

### Datasets based on cell graphs and non-cell graphs

For this work, we utilized three cell graph-based datasets: one from our previous paper on tuberculosis (TB)[50], another dataset from placenta histology[51], and lastly, the TCGA Breast Cancer Cell Classification Dataset (BRCA-M2C)[52]. The TB dataset contained 44 whole slide images (WSIs) with an average size of 42,831 x 41,159 pixels at 40X magnification. The nodes were classified into acid-fast bacilli (AFB) and the nucleus of activated macrophages. The approach used to determine the cell locations and classify the cell types is detailed in our previous work[50]. We used 34 WSIs for training and validation, while 10 WSIs were reserved for the test set. The train and test WSIs used in this study differed from those proposed in[50]. The training set had 90878 nodes, the validation set had 22708 nodes, and the test set had 76316 nodes.

The placenta dataset consisted of two cell graphs constructed from two placenta histology WSIs, combined into a single graph with nine classes. We utilized the original 64-dimensional feature set provided with the dataset for our analysis. These features primarily focussed on the morphological characteristics of the cells. Our goal was to evaluate the efficacy of knowledge distillation with cell graph datasets where the cell graph features were not included in the training process. The process of feature extraction is described in[51]. Additionally, we followed the dataset's original train, validation, and test split (considering only labeled nodes).

The BRCA-M2C dataset (Breast Cancer Dataset)[52] provided dot annotations for multi-class cell classification in breast cancer images, including the annotated cells' coordinates and corresponding labels. The cell extraction and labeling process can be found in[52]. These images were patches extracted from 1000x1000 pixels at the highest resolution and downsampled to 20x. All images were around 500x500 pixels. The cell classes included lymphocytes, breast cancer cells, and stromal cells. There were 80 image data (coordinates of the annotated cells along with their corresponding labels) under the training set, 10 image data under the validation set, and the test set consisted of 30 image data. We combined training and validation data while keeping the test data unchanged. This resulted in 19602 training nodes, 2178 validation set nodes, and 8858 test set nodes.

To determine the generalizability of our approach to non-cell graph-based datasets and in the absence of features extracted from cell graphs, we used three non-cell graph-based datasets: CoauthorCS, CoauthorPhysics and a synthetic dataset. These datasets consisted of a single graph. The CoauthorCS dataset consisted of 18,333 nodes and 163,788 edges, with nodes divided into 15 classes. A 6,805-dimensional feature vector represented each node. The training set had 12833 nodes, the validation set had 3666 nodes, and the test set had 1834 nodes. Similarly, the CoauthorPhysics dataset contained 34,493 nodes and 495,924 edges, with nodes categorized into five classes. Node features in this dataset were 8,415-dimensional vectors. The training set had 24145 nodes, the validation set had 6898 nodes, and the test set had 3450 nodes. These datasets were only used to evaluate the applicability of our approach to non-cell graph settings and were not included in ablation studies. We generated a synthetic dataset of 60,000 nodes using the preferential attachment mechanism of the Barabási-Albert model[53]. Seven topological features were extracted for this graph to represent its structural properties. The dataset training set contained 42,000 nodes, 12,000 nodes were present in the validation set, and 6,000 nodes were present in the test set, respectively.

Generally, datasets with a minority class proportion between 20% and 40% are considered to have mild imbalance, those with proportions from 1% to 20% are categorized as moderately imbalanced, and datasets with a minority class proportion of less than 1% are considered extremely imbalanced[54]. Based on this classification, TB and Breast cancer datasets had a mild imbalance. The Placenta, CoauthorCS, and Synthetic datasets demonstrated extreme class imbalance. The CoauthorPhysics dataset had a moderate imbalance.

### Construction of cell graph

Edge construction in cell graphs estimates the biological likelihood that neighboring cells interact within the same structure. The edge threshold for intercellular communication is critical in cellular studies, and many investigations have aimed to determine the optimal distance for accurately modeling these interactions.

| Node 'u' | Node 'v' | Distance 'd' in pixels |
|---|---|---|
| AFB | AFB | 615 |
| AFB | Macrophage Nucleus | 2049 |
| Macrophage Nucleus | AFB | 2049 |
| Macrophage Nucleus | Macrophage Nucleus | 2049 |

**Table 1**. Distance thresholds.

| Cell_1 | Cell_2 | k-value |
|---|---|---|
| Lymphocyte | Lymphocyte | 5 |
| Breast cancer cell | Breast Cancer Cell | 2 |
| Stromal | Stromal | 8 |
| Lymphocyte | Breast cancer cell | 5 |
| Lymphocyte | Stromal | 10 |
| Breast cancer cell | Stromal | 10 |

**Table 2**. k-values for different types of cell interactions.

Pathologists' input provides valuable guidance to refine graph representations and ensure they accurately reflect the biological relationships between cells[55]. Many prior works have employed a single threshold value to map cell-cell interactions[23,33], while some have experimented with varying edge thresholds, such as 60, 75, and 90 $\mu$ m, to identify an appropriate threshold value[56]. In contrast, our approach uses distinct threshold values for each cell-cell pair,

In the TB dataset, nodes represent either AFBs or the nucleus of activated macrophages. Edge thresholds were based upon the length of cords of the *M.tb* infected cells after 72 hours of infection[9] and the fact that macrophages can extend their pseudopods beyond their normal boundary (radius) to detect other cells farther away. We hypothesize that AFBs can interact with other AFBs within a distance of 150 $\mu$m, equivalent to 615 pixels at the magnification used in this study[57]. Likewise, activated macrophage nuclei may interact with both AFBs and each other if they are within 500 $\mu$m (2049 pixels)[10]. Our domain expert has thoroughly reviewed and validated these threshold values.

The adjacency matrix is computed as follows:

$$A_{ij} \begin{cases} 1 & \text{if } Distance(u,v) < d \\ 0 & \text{otherwise.} \end{cases}$$

Distance denotes euclidean distance computing using the equation 1. The coordinates $(x_u, y_u)$ belongs to node 'u' and the coordinates $(x_v, y_v)$ belons to node 'v' in the image.

$$d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} \tag{1}$$

The distance threshold values are tabulated in the Table 1.

For the placenta dataset, the authors utilized the intersection of the K-nearest neighbors (KNN)[58] and Delaunay triangulation[59] graphs with a k-value of 5 to generate the cell graphs. In this graph, the nodes represented cells, and the edges depicted their interactions.

For the BRCA-M2C dataset, we constructed cell graphs where nodes represent cells and edges represent interactions based on the k-nearest neighbors (KNN)[58] approach. Different k-values were used for each pair of cell types to reflect the biological significance of their interactions. The values used are tabulated in the Table 2. The adjacency matrix is calculated using the Eq. 2. The chosen k values were determined based on the cohesiveness of tumor cells and the solitary nature of stromal cells in tumors. Similarly, lymphocyte interactions were assigned moderate k values to reflect their intermediate proximity during immune surveillance, whether with tumor cells or among themselves. Figure 1 illustrates the cell graphs for various datasets.

$$A[i,j] = \begin{cases} 1 & \text{if } j \in KNN(i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

*Are all these edges required?*

While the cell graphs used in our study are generated by considering biological interactions, we acknowledge that they might not represent the optimal cell graphs. The edges in these graphs capture critical intercellular interactions. However, determining the optimal edges for such graphs remains an open research question. These interactions prove to be highly beneficial, particularly when the test set originates from a distribution different from the training set. Randomly removing edges from the cell graphs has been shown to hamper the teacher model's performance. This, in turn, degrades the performance of the student models, as the quality of the

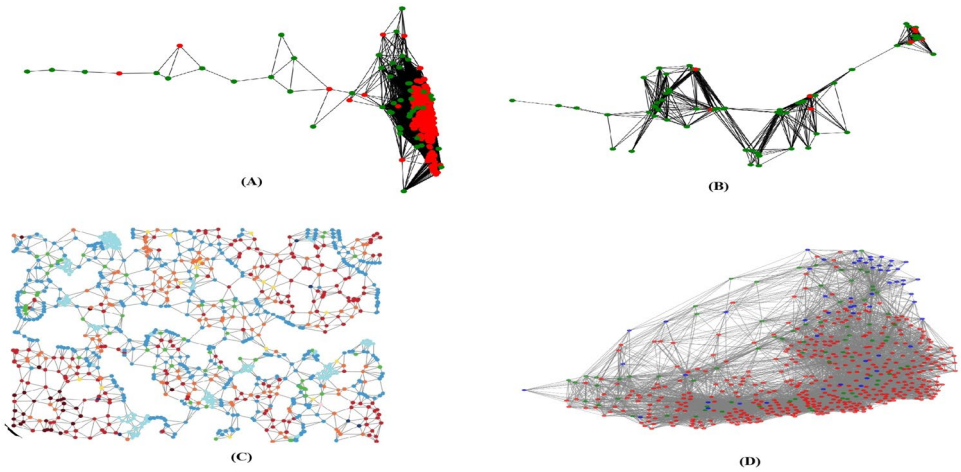


**Figure 1**. Cell graphs of the TB and BRCA-M2C datasets were generated using the NetworkX library[60] (version 3.4.2, https://networkx.org/). (**A**) Cell Graph generated for a TB image. Acid-fast bacilli (AFB) cells are shown in red, and the nucleus of activated macrophages is depicted in blue. Black edges represent interactions. (**B**) Cell Graph generated for a normal lung tissue, i.e., not infected. (**C**) Cell Graph acquired from the Vanea et al.[51], licensed under Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/). (**D**) Cell Graph generated from the BRCA-M2C dataset, where red nodes represent lymphocytes, blue nodes represent tumor cells, green nodes represent stromal cells, and gray edges denote their interactions, created using different k-values for specific cell interactions.

| Feature names | Features (feature number) |
|---|---|
| Graph Features | Eccentricity (1), Closeness_of_node (2), Average_Clustering (3), Node_Clustering (4), Sørensen (5), Salton (6), Hub_Promoted (7), Hub_Depressed (8), Centrality (9), Mean_all_neighbors (10), Skew_all_neighbors (11), Kurtosis_all_neighbors (12) |
| Morphological Features | X (13),Y (14), Contrast (15), Energy (16), Correlation (17), Homogeneity (18), ASM_value (19), Dissimilarity (20), Variance (21), Mean_Image (22), Standard_Deviation (23), Area (24), Major_Axis (25), Minor_Axis (26), Eccentricity_object (27), Perimeter (28), Diameter_object (29), Circularity (30), Mean_convex_hull (31), SD_convex_hull (32) |

**Table 3**. Features. Note: Skew_all_neighbors and Kurtosis_all_neighbors are computed based on the distribution of edge lengths between neighboring nodes.

teacher's logits diminishes. The concept of optimal cell graphs with the right amount of connectivity to balance model complexity and performance remains an emerging area of research that requires further exploration.

## Feature extraction

We tested the efficacy of our approach under different feature sets across datasets. We combined local cell graph features with morphological features for the TB dataset. For the Placenta dataset, we used only morphological features (along with inherent variations in cell appearance). For the BRCA-M2C dataset, we utilized only the local cell graph features. For the Coauthorship datasets, we did not extract additional features. Instead, we used the existing original features provided by the datasets.

*TB dataset*

In[50], combining morphological and graph features resulted in the best results for CG-JKNN. Hence, we use this combination to train our models in this work. Table 3 denotes the extracted features; the description can be found in the paper that introduced it.

*Placenta dataset*

For the placenta dataset, we used the features defined in the original paper. Specifically, the node features are defined using the nucleus coordinates as node coordinates and the 64-dimensional embeddings from the penultimate layer of the cell classifier model. These features primarily encode morphological information about cells rather than cell graph structural information.

*BRCA-M2C dataset*

For the BRCA-M2C dataset, we extracted the local graph features from the cell graphs generated. The extracted features are listed in Table 4.

| Feature names | Features (feature number) |
|---|---|
| Graph features | Degree (1), Betweenness_centrality (2) , Eccentricity (3), Closeness_of_node (4), Node_Clustering (5), Kurtosis_all_neighbors (6), Mean_all_neighbors (7), Skew_all_neighbors (8), Sørensen (9), Salton (10), Hub_Promoted (11), Hub_Depressed (12) |

**Table 4**. Features.

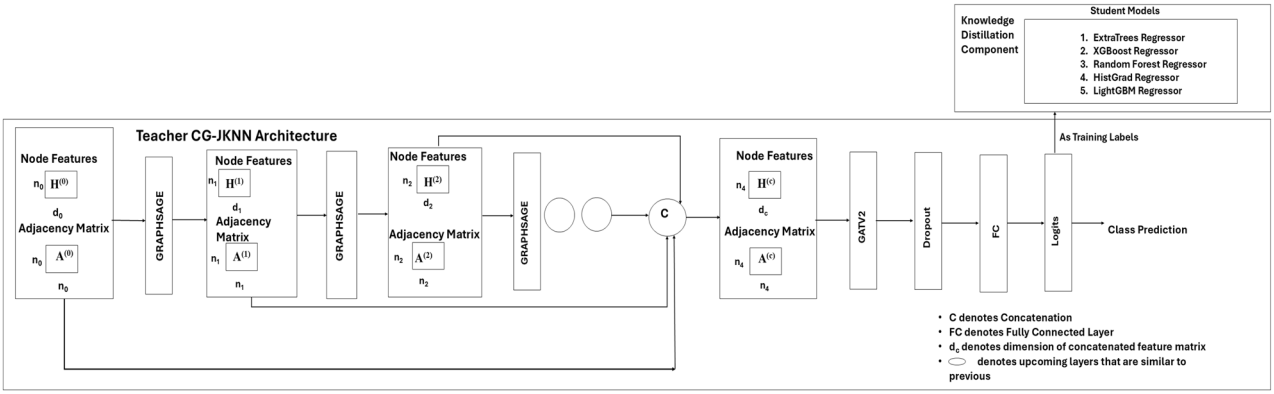


**Figure 2**. Architecture of the teacher model used for knowledge distillation. To obtain the temperature-scaled logits, as discussed in the ablation study, a temperature-scaling block needs to be incorporated between the logits generated by the teacher model and the input to the student models.

## Distilling the knowledge from CG-JKNN (teacher) to tree-based ensembles (students)

Based on the CG-JKNN architecture, the teacher model is designed for node-level classification tasks. A graph is defined as $G = (V, E)$, where $V$ denotes the set of nodes, and each node $v$ is associated with a $d$-dimensional feature vector $x_v \in \mathbb{R}^d$. The edges $E$ are represented by $e_{u,v} = (u, v)$, indicating a connection between nodes $u$ and $v$. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ encodes the graph structure.

The architecture of our teacher model and the flow of our proposed work are depicted in Fig. 2. To train the teacher GNN, we utilize cell graphs $G$ constructed along with their associated node features $x_v$. During the training phase, the model learns to classify each node by predicting its label based on the provided labeled graphs. During testing, the trained GNN receives unseen cell graphs $G$ and their associated node features $x_v$. The model predicts the test node labels, which are then compared against the true labels in the test set to evaluate performance.

Each node's hidden features $h_v^{(l)} \in \mathbb{R}^d$ in the $l$-th layer are initialized with the input features as $h_v^{(0)} = x_v$. The GraphSAGE layers process node representations, employing a mean aggregation function as shown in Eq. 3 to gather information from neighboring nodes. In our previous work[50], we experimented with both mean and max aggregators and found the mean aggregator to achieve superior performance consistently. This also aligned with prior studies that demonstrate the effectiveness of mean aggregation in node classification tasks[61,62]. Therefore, we selected the mean aggregator.

$$\boldsymbol{h}_{\boldsymbol{N(v)}}^{(\boldsymbol{l})} = \text{MEAN}\left(\left\{\boldsymbol{h}_{\boldsymbol{u}}^{(\boldsymbol{l-1})}, \forall u \in N(v)\right\}\right) \tag{3}$$

Here, $h_{N(v)}^{(l)}$ represents the aggregated neighborhood representation, and $h_u^{(l-1)}$ corresponds to the representation of neighboring node $u$ from the previous layer. The node's updated representation is computed using Eq. 4.

$$\boldsymbol{h}_{\boldsymbol{v}}^{(\boldsymbol{l})} = \sigma\left(W \cdot \left[\boldsymbol{h}_{\boldsymbol{v}}^{(\boldsymbol{l-1})}, \boldsymbol{h}_{\boldsymbol{N(v)}}^{(\boldsymbol{l})}\right]\right) \tag{4}$$

Here, $W$ is the learnable weight matrix, and $\sigma$ denotes the activation function (ReLU).

The "jumping knowledge representation learning" mechanism[12] is incorporated to combine multi-layer node representations. This approach concatenates representations from all layers to form a comprehensive node representation (Eq. 5) instead of using only the final layer's representation. The authors in[12] explored three different aggregation mechanisms: concatenation, max-pooling, and an LSTM-based attention mechanism. Our network adopts the concatenation-based jumping knowledge mechanism for aggregating node representations.

$$\boldsymbol{h}_{\boldsymbol{v}}^{(\boldsymbol{Concatenated})} = \text{Concatenate}\left[\boldsymbol{h}_{\boldsymbol{v}}^{(\boldsymbol{1})}, \ldots, \boldsymbol{h}_{\boldsymbol{v}}^{(\boldsymbol{l})}\right] \tag{5}$$

After concatenation, the node representations are passed through a GATv2 layer[63], which refines the representations using an attention mechanism. The attention coefficients $\alpha_{vu}$ are computed as:

$$\alpha_{vu} = \text{softmax}_u\left(\text{LeakyReLU}\left(\boldsymbol{a}^T\left[W\boldsymbol{h}_{\boldsymbol{v}}^{(\text{Concatenated})} \quad \| W\boldsymbol{h}_{\boldsymbol{u}}^{(\text{Concatenated})}\right]\right)\right) \tag{6}$$

Finally, the node representations are updated as shown in Eq. 7, Later, the softmax function applied to obtain the class probabilities.

$$\boldsymbol{h}_{\boldsymbol{v}}^{(\boldsymbol{GAT})} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} W\boldsymbol{h}_{\boldsymbol{u}}^{(\text{Concatenated})}\right) \tag{7}$$

Here, $\mathcal{N}(v)$ denotes the neighbors of node $v$, and $\sigma$ is the activation function. We use a rectified linear unit (ReLU) as the activation function. Over-smoothing is a critical issue in GNNs. It arises when deep networks cause node features to converge, losing their distinctiveness. Existing approaches address this challenge using various strategies. Energetic Graph Neural Networks employ energy-based modeling[64], while Graph DropConnect introduces graph-specific dropout[65]. Graph-coupled oscillator Networks use non-linear oscillators to modify GNN dynamics[66], and residual connections improve the information flow in deep GNNs to counter over-smoothing[67]. For this study, we adopted the DropEdge technique[68]. It mitigates over-smoothing by randomly removing a proportion of edges during training. Using the edge index representation for graph connections, we experimented with various dropping rates.

Logits represent the unnormalized outputs of the model. It provides richer information compared to class probabilities. It has been shown in the literature that training the student model directly on the logits allows for more effective learning of the internal representations captured by the teacher[18]. This approach enables the student to mimic the teacher's learned patterns better. Additionally, it helps avoid the information loss that typically occurs when logits are transformed into probabilities. Hence, we extract the logits before applying the softmax function for knowledge distillation and use them as labels to train the student regressor models.

In general, the KD loss[69] is formulated to align the predictions of the student model with those of the teacher model by minimizing the divergence between their output distributions. This is typically achieved by leveraging the Kullback-Leibler (KL) divergence. While this approach is effective for neural network-based student models that undergo continuous updates during training, it is not directly applicable to our scenario. In our study, the student models are tree-based ensembles that do not rely on iterative gradient updates. As a result, we don't utilize this loss function.

After training on the teacher's logits as targets, the student models generate predictions, which are converted into probabilities using the softmax function. These probabilities are evaluated to calculate performance metrics such as accuracy and F1-score. We specifically chose non-linear models for students because the teacher logits, serving as labels, are inherently non-linear. For the student models to effectively learn from these logits, they must possess sufficient capacity (or complexity) to capture the underlying non-linear relationships embedded in the teacher's predictions. We employ tree-based ensemble regressors as student models, as described in the Table 5. For brevity, we will often refer to these models by their specific names rather than repeatedly using the term 'regressor' throughout the paper.

## Estimating the complexity of tree-based ensemble models-an approximation and distillation quality score

Understanding the complexity of student models is essential to evaluating the quality of knowledge distilled from the teacher model. Black-box models, including various ensemble techniques, diverge from traditional likelihood-based frameworks and present challenges in directly assessing model complexity. This is mainly because the number of parameters in such models does not accurately represent their degrees of freedom. The concept of Generalized Degrees of Freedom (GDF), introduced by Ye[70] and later applied to machine learning by Elder[71], serves as a metric for assessing the complexity of models. For instance, in the case of a two-dimensional decision tree scenario, Elder[71] has observed that combining multiple trees through bagging leads to an ensemble with a Generalized Degrees of Freedom (GDF) complexity that is lower than that of any single tree within the ensemble. In[72], they employed GDF to estimate the number of parameters for the random forest model that was

| Student model | Description |
|---|---|
| ExtraTrees | An ensemble model that combines multiple randomized decision trees for regression. |
| XGBoost | A gradient boosting model that optimizes performance using weak learners. |
| HistGradientBoost | A histogram-based gradient boosting regressor for efficient training on large datasets. |
| Random Forest | An ensemble method that uses multiple decision trees for robust regression. |
| LightGBM | A gradient boosting framework optimized for speed and efficiency with large data. |

**Table 5**. Student models and their descriptions.

utilized to predict cell-type specific enhancer-promoter interactions by leveraging the information of protein-protein interactions between transcription factors.

Despite the utility of GDF in providing an estimate of model complexity, it has some challenges. Firstly, the sensitivity of GDF to perturbations in the data means that the degree to which GDF reacts can vary significantly depending on the specific modeling approach being used. This variability indicates that a GDF estimation method that works for one model type may not be suitable for another. In addition, the absence of a robust, universally applicable method for estimating GDF complicates its implementation across different data distributions and model architectures. These drawbacks highlight the complexity of accurately assessing model behavior in machine learning and the need for further research in developing more adaptable metrics like GDF[73].

A standard metric for choosing models is the Akaike Information Criterion (AIC)[74], which illustrates the trade-off between model complexity and goodness of fit. Models with reduced AIC values indicate a better balance between the model complexity and goodness of fit. It is computed using the Eq. 8. $M_k$ denotes the model with dimension k. $L(M_k)$ is the likelihood corresponding to the model $M_k$

$$\mathrm{AIC}\left(M_k\right) = -2\log L\left(M_k\right) + 2k \tag{8}$$

However, one limitation of the Akaike Information Criterion (AIC) is its unsuitability for non-parametric model selection[75]. Models such as Random Forest are non-parametric[76]. It is a common misconception that non-parametric models have no parameters. They can be thought of as having an infinite number of parameters. This characteristic suggests that the complexity of non-parametric models can grow to capture increasingly precise information within the data as the number of data rises[76]. Few papers have computed the AIC for models such as Random Forest in[77]. This study developed a machine-learning model to simulate the effect of masks on motor sound, utilizing noise level data in decibels from various operation frequencies of motors at the National Synchrotron Radiation Research Center (NSRRC). Three group indicators were used to assess the learning performance: the Akaike Information Criterion (AIC), the Hannan-Quinn Information Criterion (HQIC), the Schwartz-Bayesian Criterion (SBIC), and the Akaike Information Criterion with Small Sample Correction (AICc). However, based on the information provided, the specific method used to determine the number of parameters ('k') for the AIC score is unclear.

When models are estimated using maximum likelihood, the choice of model based on minimizing the cross-validation error leads to asymptotically equivalent decisions as selecting the model that minimizes the AIC[78]. Based on this, the authors in[73] argued that it should be possible to extract a measurement from $l_{CV}$ (which denotes the sum over K folds of the log-likelihood of the validation subset that estimates model complexity). The equation in 9 denotes the asymptotic equivalence between AIC and leave one out cross validation (LOOCV). Based on this, the number of parameters p can be estimated using the Eq. (11). $l_m$ denotes the maximum log-likelihood of the original (non-cross-validated) model, and $l_{CV}$ represents the sum over K folds of the log-likelihood of the validation fold.

$$\mathrm{AIC} = -2\ell_m + 2\hat{p} \approx -2\ell_{\mathrm{CV}} \tag{9}$$

$$\begin{aligned} -2lm + 2p &\approx -2lcv \\ 2p &\approx -2lcv + 2lm \\ p &\approx 2(lm - lcv)/2 \end{aligned} \tag{10}$$

$$\hat{p} \approx \ell_m - \ell_{\mathrm{CV}} \tag{11}$$

In our work, we have employed tree-based ensemble regressors as student models. These are non-likelihood models. In[73], the authors found the notion of applying GDFs to non-likelihood models to improve information-theoretic metrics of model fit (like AIC) was associated with the high cost of processing and produced inconsistent results. While cross-validation was a more direct method, it was less stable than GDFs. To determine the model complexity metric, they suggested repeated 10-fold cross-validation. Cross-validation is suitable for models that do not make likelihood assumptions since it can but need not, use the likelihood fit.

We build our methodology based on this idea. We utilize the sum of squared errors (SSE) to approximate the log-likelihood term. It suits our models that do not directly maximize the likelihood function. A higher maximum log-likelihood value indicates that the observed data is more probable under the model, which is interpreted as a better fit. A lower SSE suggests that the model's predictions are closer to the actual observed values, which is also interpreted as a better fit.

Equation (12) shows the computation of model complexity with SSE. The $SSE_{full}$ denotes the sum of squared errors on the training set, and $SSE_{CV}$ denotes the SSE of the cross-validation. The logarithm helps to scale and normalize the SSE in relation to the number of observations 'n.' In our experiments, we implemented a trial of 10-fold cross-validation recognizing the expensive computational demands of LOOCV. However, it does introduce some level of Monte-Carlo variability, resulting from not averaging all possible leave-one-out sets, as would be the case with LOOCV[73]. We observed slight variations in these estimates across different runs during our experiments. To ensure stable and reliable estimates, we recommend future researchers to conduct multiple runs, as suggested in[73].

$$\hat{p} \approx n/2\ln\left(\frac{SSE_{CV}}{n}\right) - n/2\ln\left(\frac{SSE_{full}}{n}\right) \tag{12}$$

These terms capture the fit by indicating how close the model's predictions are to the actual data points, with the logarithm helping to scale and normalize the SSE in relation to the number of observations. The supplementary files provide additional results on how model complexity changes under varying parameters. Henceforth, the term 'number of parameters' for non-neural models in this study will denote the effective complexity,$\hat{p}$.

Based on the complexity approximated, we compute the distillation quality metric, which measures the effectiveness of the distillation process. Inspired by[79], we employ a slightly modified version of the distillation quality metric to evaluate the performance of various student models. Its computation is shown in equation 13. Instead of using accuracy, we use a weighted F1 score in our metric when dealing with imbalanced datasets.

$$DS = \alpha \cdot \left(\frac{\text{student}_c}{\text{teacher}_c}\right) + (1-\alpha) \cdot \max\left(0, 1 - \frac{\text{student}_{f1}}{\text{teacher}_{f1}}\right) \tag{13}$$

$\text{student}_c$ and $\text{teacher}_c$ denote their respective complexities (in terms of parameters), and $\text{student}_{f1}$ and $\text{teacher}_{f1}$ denote their F1-scores (weighted). The approach of computing the number of parameters of our student models is described under section "Estimating the complexity of tree-based ensemble models-an approximation and distillation quality score". The second term incorporates the max function to handle cases where the student outperforms the teacher. The authors in[79] emphasize that the choice of the parameter $\alpha$ is left to the designers, allowing them to prioritize either model size or accuracy according to their system's requirements. For instance, a value of $\alpha > 0.5$ would be appropriate if smaller model sizes are more critical. In our work, to balance the importance of model size and performance, we set $\alpha = 0.5$, giving equal weight to these two factors. For balanced datasets, accuracy can be used instead of F1-scores to evaluate performance. In cases where the student outperforms the teacher, the ratio of student performance to teacher performance exceeds one. To address this, we have adjusted the score to ensure it remains non-negative. In our approach, a score of zero is achieved when the student model outperforms the teacher while maintaining a much smaller size than its teacher.

## Ablation studies

We conducted three ablation studies, primarily focusing on cell graph data sets. The first study explored training with ensembled logits from the teacher and the best-performing student model. The second study aimed to analyze the differences in the importance of features when the models were trained using teacher logits compared to when they were trained using hard labels. The third study compares the effectiveness of transferring teacher knowledge via distillation into two types of student models: an Artificial Neural Network (ANN) and non-neural models.

*Combining teacher and top student: ensemble model training*

The goal of knowledge distillation from several teachers is to produce a good student who inherits the majority of the ensemble's performance without raising the computational cost of inference. First, building highly predictive teacher ensembles is required to produce strong student models with distillation[80]. A few works focus on ensemble distillation on unlabeled datasets[81–83]. Since our study focuses on labeled data, we explicitly evaluate approaches relevant to labeled datasets for our distillation process, where the crucial problem is how to assign different weights to individual teachers within the ensemble[81]. In[84], they proposed an ensemble model that unified three distinct knowledge distillation methods–feature-based, response-based, and relation-based on the CIFAR-10 and CIFAR-100 benchmarks. The distillation utilized a lightweight ResNet-20 student model with 0.27 million parameters and a ResNet-110 teacher model with 1.7 million parameters. The authors in[85] trained an ensemble of various Multi-Task Deep Neural Networks (MT-DNNs (teachers)), achieving superior performance over any single model. Subsequently, they trained a single MT-DNN (student) through multi-task learning, effectively distilling knowledge from the ensemble of teachers. Wang et al.[86] trained one segmentation teacher CNN on synthetic samples with accurately known ground truth fault labels and another classification teacher CNN on field samples with manually annotated labels. Following this, a classification student network was trained on samples created by aggregating the predictions from both teacher models through a voting mechanism. The authors in[87] proposed MT-BERT, a novel approach to multi-teacher knowledge distillation focused on the compression of pre-trained language models. They devised a co-finetuning framework that simultaneously fine-tuned multiple teacher models employing a unified pooling and prediction module to align their output hidden states. This methodology enhanced the collaborative teaching of the student model. Chebotar and Waters[88] discovered an effective ensemble of acoustic models comprising LSTM and CLDNN architectures developed with diverse training objectives, where the student model was a CLDNN. Initially, the research involved identifying the optimal fixed weights for merging the outputs of teacher models to maximize accuracy. The knowledge was later distilled into the student model using the soft labels generated by the ensemble. The authors in[89] proposed a dynamic weighting approach for each teacher, demonstrating its effectiveness in logits-based and feature-based distillation through extensive experiments. They treated the process as a multi-objective optimization problem to find a more effective training direction.

For this ablation study, we consider both the CG-JKNN and the highest-performing student model as teacher models to investigate their combined impact on knowledge distillation. We adopt the methodology proposed in[88], which involves identifying optimal fixed weights for merging the outputs of teacher models to maximize the F1 score on the validation set. Following this, we distill a student model from the ensemble output generated through this optimized combination. Equation (14) illustrates the method for aggregating outputs from the teacher GNN and LightGBM models. The detailed approach is shown in the algorithm 1.

$$L_{\text{ensemble}}(x) = w_{\text{gnn}} \cdot L_{\text{gnn}}(x) + w_{\text{lightgbm}} \cdot L_{\text{lightgbm}}(x) \tag{14}$$

$L_{\text{ensemble}}(x)$ is the ensembled output for a given input $x$. $L_{gnn}(x)$ is the logit output from the GNN model for a given input $x$. $L_{\text{lightgbm}}(x)$ is the raw decision score output from the LightGBM model for a given input $x$. $w_{\text{gnn}}$ and $w_{\text{lightgbm}}$ are the weights applied to the outputs from the GNN and LightGBM models, respectively. It can be adapted to incorporate the outputs of other high-performing student models.

Input: *Logits_GNN*, logits from the teacher GNN model
Input: *Predictions_Best_Student*, Raw decision scores obtained on training set (before softmax)/predictions from the best student model
Output: *best_weights*, optimal weights to combine outputs from GNN and best student model
Initialize *best_val_f1_score* ← 0
Initialize *best_weights* ← *None*
**for** each weight $w$ in $[0, 0.1, \ldots, 1]$ **do**
  Set *weight_gnn* ← $w$ and *weight_best_student* ← $1 - w$
  Initialize *predictions_val* ← []
  Normalize *Logits_GNN* and *Predictions_Best_Student* if they are on different scales
  Compute combined output: *final_outputs* ← *weight_gnn* × *logits_gnn* + *weight_best_student* × *predictions_best_student*
  **for** each column $i$ in *final_outputs* **do**
    Instantiate model
    Fit model on training data with $i^{th}$ output as target
    Predict on validation data and append it to *predictions_val*
  **end for**
  Evaluate performance on validation data after converting the predictions in *predictions_val* to probabilities using softmax transformation
  **if** performance > *best_val_f1_score* **then**
    Update *best_val_f1_score* and *best_weights* with current values
  **end if**
**end for**

**Algorithm 1**. Optimal Weight Finding for Ensemble of Teacher GNN and Best Student model

*Feature importance: comparing students trained with and without teacher guidance*
We aimed to analyze the differences in feature importance of student models trained on teacher logits and their counterpart trained on hard labels. Literature suggests that students trained on logits are better equipped to mimic the behavior of the teacher model[18]. Thus, this analysis can also serve as an approach to explore how a student model trained on logits may partially act as a proxy for interpreting the teacher's decision-making process. For this experiment, we selected the student model that performed best on the held-out test set. To determine feature importance, we utilized the "feature importances" attribute of the model. Additionally, to assess how some of these important features contribute to predictions for each class and the direction of their impact, we employed SHapley Additive exPlanations (SHAP) plots[90]. Our objective was not to compare these techniques but to leverage SHAP for a deeper understanding of how features influence model predictions. In future work, we plan to incorporate advanced techniques such as permutation-based methods (e.g., Boruta importance)[91] and knockoff approaches[92], as these methods provide a more robust and accurate assessment of a feature's predictive abilities within a model[93].

It is important to note that the student model can act as an interpretable approximation of the teacher by reflecting its emphasis on certain cell graph level or morphological features. However, it cannot leverage the graph structure and complex node relationships that the teacher model captures through message passing. Instead, the student operates solely on feature values and the logits provided by the teacher. It thus limits its ability to fully replicate the teacher's reasoning process.

*Comparing effectiveness of knowledge distillation into ANN vs. non-neural student models*
In this ablation study, we selected an ANN as the neural student to ensure both model types rely solely on the features and implicit relational knowledge provided through the logits of the teacher GNN. This avoids the additional advantage of directly exploiting cell graph structures that a GNN would have and ensures that any observed differences in performance stem directly from the effectiveness of the distillation process.

We designed a shallow network with one hidden layer to maintain a smaller student model and its structure is illustrated in Fig. 3. The hyperparameters, such as hidden dimensions, alpha (which balances the two losses), and learning rate, were optimized using Optuna over 50 trials, selecting those that maximized the validation F1 score. We also constrained the hyperparameter search space to ensure that the ANN model parameters remained comparable to those of the non-neural student models.

Hinton et al.[15] discovered that the effectiveness of the student model's learning process is significantly enhanced when it is trained using both the soft target provided by the teacher model and the actual ground

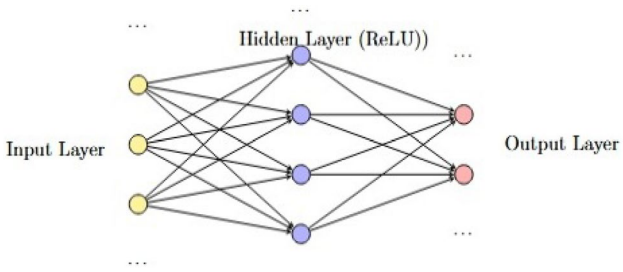


**Figure 3**. Architecture of our shallow ANN student model. The ellipses denote that additional neurons are present in the layer but are not explicitly illustrated for clarity.

truth. This approach involves a combined loss function that integrates two key components: the traditional cross-entropy loss and a knowledge distillation-specific loss term.

The overall loss function for knowledge distillation can be expressed as shown in the equation 15.

$$L_{KD} = \alpha L_{CE}\left(p_s, y\right) + (1-\alpha)\tau^2 KL\left(p_s^\tau, p_t^\tau\right) \tag{15}$$

Here, $L_{CE}\left(p_s, y\right)$ represents the cross-entropy loss. The second component, $\tau^2 KL\left(p_s^\tau, p_t^\tau\right)$, is the knowledge distillation term. $p_s^\tau$ and $p_t^\tau$ denote the softened outputs of the student and teacher models, respectively, after applying the temperature scaling with parameter $\tau$. KL stands for the Kullback-Leibler divergence, a measure of how one probability distribution diverges from a second, reference probability distribution. $\alpha$ is a hyperparameter that controls the balance between the traditional cross-entropy loss and the knowledge distillation loss. In our work, we observed that logits before calibration already produced good results, and consequently, we set the temperature $\tau$=1.

Hinton et al.[15] suggested using a weighted average between the distillation loss and the student loss by setting $\beta = 1 - \alpha$, and in one of their experiments, they used $\alpha = \beta = 0.5$. Other works that utilize knowledge distillation treat this weight as a tunable parameter[94–96]. In our work, we treat the weight parameter $\alpha$ as a hyperparameter. Additionally, we present results using a fixed $\alpha$ value of 0.5.

## Generalizability of knowledge distillation under various dataset complexities

To investigate whether all models benefit from knowledge distillation and assess the effectiveness of our approach across various dataset complexities, we conducted experiments on multiple datasets (cell graph and non-cell graph). These datasets presented challenges, such as distribution shifts, and structural complexities in training and testing graphs. Importantly, for Coauthorship datasets, we did not extract local graph features but instead utilized the original dataset features. This allowed us to test the efficacy of knowledge distillation in the absence of graph-specific features. The logits obtained from GNN trained on these coauthor networks could encapsulate rich information by reflecting relationships between node features (keywords) and the graph structure (Coauthorship network). For instance, if an author is involved in interdisciplinary work, their logits may encode soft probabilities across multiple fields, capturing the uncertainty or overlap between class labels.

*Graph complexity*

We hypothesize that for knowledge distillation to be effective when the teacher is a GNN learning from the graph, the graph must possess sufficient complexity. In such cases, the logits transferred from the GNN provide valuable information that student models can leverage.

According to the literature, graph complexity measures can be categorized into deterministic and probabilistic methods[97]. Deterministic approaches include Kolmogorov complexity, substructure counting, and generative models. Probabilistic methods involve entropy functions (such as Shannon's entropy) applied to probability distributions over graph structures with intrinsic and extrinsic subcategories. In our work, we focus on graph energy, a concept originating from molecular and quantum chemistry, as a metric to evaluate how graph structural complexities affect knowledge transfer from a teacher GNN to student models[98,99]. It is computed using the Eq. (16).

$$C = \left(\frac{1}{|A|}\sum_{k=1}^{|A|} b_k\right)\sum \mathrm{SVD}(M) \tag{16}$$

Here $b_k$ represents the edge weights if any, $|A|$ denotes the number of edges in the graph, and $\mathrm{SVD}(M)$ is a vector of singular values of the matrix $M$[98].

*Distribution shift in the data*

The distribution shift[100–102] can be broadly categorized into three types: Covariate shift, label shift, and concept shift. The feature distribution changes in the covariate shift case, while the label distribution does not. On the other hand, label shift happens when the distribution of the labels varies while the feature distribution remains the same. Concept shift, also called conceptual drift, arises when the actual relationship between the inputs and

labels evolves, reflecting a change in the underlying concept the model is attempting to capture. There exist multiple ways to detect covariate shifts. We can compare summary statistics or employ dissimilarity measures like Earth mover's distance. For statistical rigor, hypothesis tests such as the Kolmogorov-Smirnov or Chi-squared tests are used to determine significant distributional differences[103].

For this work, we utilized Kernel Principal Components Analysis (Kernel PCA) for dimensionality reduction, selecting the number of components that captured above 95% of the dataset's variance. Subsequent univariate Kolmogorov-Smirnov tests, with Bonferroni correction[104] applied to an alpha of 0.01, rigorously adjusted our significance levels to control the cumulative Type I error rate across multiple hypotheses. The mean of all significant KS statistics was computed to summarize the extent of covariate shift across the K dimensions. Moreover, for the computationally expensive TB and Placenta dataset, we subsampled 20,000 points to ensure the feasibility of the analysis while maintaining the representativeness of the original data. The mean KS statistic calculated may not fully reflect the entire degree of shift in the dataset. However, our primary goal was to demonstrate the presence of a shift.

To determine the covariate shift in non-cell graph-based datasets, we calculated the percentage of features with covariate shift by performing univariate KS tests directly on the scaled features. This was due to the high dimensionality of the dataset, as the large number of components required to achieve 95% variance capture would have made our initially proposed approach computationally expensive. For label shift detection, we employed the Chi-squared test[105] to evaluate the consistency of class distributions between the different data subsets. This involved constructing a contingency table based on the frequency counts of each unique class in these subsets. After computing the Chi-squared statistic, we assessed the p-value to determine whether the observed distributional differences were statistically significant.

## Can logit calibration enhance student guidance?

Neural networks produce poorly calibrated predictions that can be either overconfident or underconfident. GNNs can be miscalibrated too[106]. Calibration primarily aims to make predicted probabilities more reliable. In our study, we were particularly interested in investigating whether logit calibration could enhance the guidance provided to our student models. It is important to note that logit calibration does not impact the performance of the teacher model itself. Previous studies[107,108] have demonstrated how calibration can impact models' accuracy and other performance metrics. Additionally, the authors in[109] introduced the concept of addressing mis-instruction through logit calibration. This work highlighted that enhancing target logits while preserving the relative proportions among non-target logits can significantly improve the utility of logits for knowledge distillation. These works primarily dealt with neural models as students. Wang et al.[110] observed that GNNs tend to be underconfident, in contrast to the majority of multi-class classifiers, which are generally overconfident. This necessitated the use of various techniques to calibrate the logits. Guo et al.[111] proposed temperature scaling to address the miscalibration issue found in modern neural networks. Kuleshov et al.[112] introduced a straightforward calibration method based on isotonic regression. Another approach was ensemble-based temperature scaling[113]. Methods such as temperature scaling preserved accuracy by maintaining the per-node logit rankings unaltered[114].

To achieve calibration, in this work, we employed isotonic regression and temperature scaling as post-hoc calibration methods. In traditional settings, isotonic regression is employed for binary classification tasks. To extend isotonic regression to multiclass scenarios, we adopt a one-vs-all strategy[115,116]. We measured the Brier score (Stratified) and negative log-likelihood before and after calibration, as they are proper scoring rules and provide a truthful measure of the accuracy of probabilistic predictions[117]. To learn the temperature T, it is considered best practice to use a validation set or perform cross-validation. We used 5-fold cross-validation (2 folds if the dataset is highly imbalanced) by splitting the training logits into train and validation folds. We learned two temperatures using the validation fold to optimize both the Brier score and the log loss. Our paper refers to the probabilities obtained after calibration using Eq. (17) as calibrated probabilities (calibrated probs). The overall score mentioned in the paper represents the mean of the scores calculated individually for each class.

$$\hat{p}_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_{j=1}^{C} \exp\left(\frac{z_j}{T}\right)} \tag{17}$$

where $\hat{p}_i$ represents the calibrated probability for class $i$, $z_i$ is the logit for class $i$ (pre-softmax output of the model), $T > 0$ is the temperature parameter learned using a validation set or cross-validation, and $C$ is the total number of classes.

## Experimental setup and hyperparameters

We implemented the models using the PyTorch framework[118] and ran them on one NVIDIA A100 GPU. The hyperparameters of the teacher model were chosen with the assistance of Optuna[119], a Python library for hyperparameter optimization. We ran 50 trials to optimize the model hyperparameters, aiming to achieve the highest weighted F1 score on the validation set for imbalanced datasets. We used the cross-entropy loss function during training when the class imbalance was mild/moderate. We utilized a weighted cross-entropy loss function for scenarios with extreme class imbalance. The teacher model was run for 80 epochs. We used an Adam optimizer. The hyperparameters of the teacher model associated with each dataset are tabulated in Table 6. The features were scaled using the standard scaler. As performance metrics, we evaluated the accuracy and weighted F1 score. The temperatures used to calibrate the logits are also presented. The first temperature minimizes the stratified Brier score, The second temperature minimizes the log loss.

| Dataset | Teacher model hyperparameters | Complexity | Temperature (brier score) | Temperature (log loss) |
|---|---|---|---|---|
| TB | Num_GraphSage layers: 25, Num_GAT layers: 1, hidden_channels: 33, lr: 0.002, weight_decay: 5e-4 ,dropout=0.1 | 107848 | NA | NA |
| Placenta | Num_GraphSage layers: 6, Num_GAT layers: 1, hidden_channels: 45, lr: 0.006440304794081112, weight_decay: 9.480520388945085e-05 ,dropout=0.1 | 49608 | 0.6931 | NA |
| BRCA-M2C | Num_GraphSage layers: 16, Num_GAT layers: 1, hidden_channels: 37, lr: 0.00253 weight_decay: 2.56e-05, dropout=0.1 | 82220 | 1.1068 | 0.98369 |
| CoauthorPhysics | Num_GraphSage layers: 9, Num_GAT layers: 1, hidden_channels: 17, lr: 0.0018, weight_decay: 2.5339174600421627e-05, dropout=0.3669 | 581359 | 0.7614 | 0.6155 |
| CoauthorCS | Num_GraphSage layers: 5, hidden_channels: 10, lr: 0.004436311854841181, weight_decay: 2.1138365253049543e-05 | 274160 | 0.5096 | NA |
| Synthetic Dataset | Num_GraphSage layers: 10, Num_GAT layers: 1, hidden_channels: 40, lr:0.003, weight_decay: 5e-4 | 56406 | 1.2 | 0.9 |

**Table 6**. Teacher model hyperparameters and temperature values for datasets.

| Dataset | Edge homophily ratio |
|---|---|
| TB | 0.6375 |
| Cell Graph 1-Placenta | 0.9868 |
| Cell Graph 2-Placenta | 0.9984 |
| BRCA-M2C | 0.2028 |
| CoauthorCS | 0.8081 |
| CoauthorPhysics | 0.9314 |
| Synthetic Dataset | 0.6630 |

**Table 7**. Edge homophily ratios.

To maintain smaller student models, we set the number of estimators in the students to 6, with the maximum depth varying between 8 and 16 (such as 8,12,16, etc) and the number of leaf nodes fixed at 50. However, we allowed the number of leaf nodes to be 300 for our complex TB dataset. The learning rate of the boosters was set to 0.3, while all other parameters were kept at their default values. The specific depths of student models are detailed in the results section corresponding to each dataset. It is important to note that the student model performances reported are specific to the chosen hyperparameter configurations. We acknowledge that the results could vary with a more extensive hyperparameter search.

The edge homophily of the graphs used is shown in the Table 7. It is the ratio that measures the proportion of edges in a graph that connect nodes of the same class label. The equation to compute edge homophily is given in 18.

$$h = \frac{|\{(u,v) : (u,v) \in \mathscr{E} \wedge y_u = y_v\}}{|\mathscr{E}|} \tag{18}$$

where: $h$ denotes the edge heterophily score, $|\mathscr{E}|$ is the total number of edges in the graph, $(u, v)$ represents an edge between nodes $u$ and $v$, $y_u$ and $y_v$ are the labels of nodes $u$ and $v$.

As stated in[120], a high edge homophily ratio indicates strong homophily where $h \to 1$ while a low edge homophily ratio indicates strong heterophily where $h \to 0$ .

## Results

### Covariate and label shift across datasets

Table 8 presents the Mean KS statistic, chi-squared statistics, and corresponding p-values for each dataset pair.

Based on the results, we observe a covariate shift in the test set of the TB dataset, as the test nodes are taken from separate graphs compared to the training and validation nodes. Additionally, the Chi-squared statistic indicates the presence of a label shift in the data. In the placenta dataset, we did not observe a large covariate shift between the validation and test sets, while a covariate shift is observed in other splits. This could be due to how the nodes were sourced. However, no label shift was detected in this dataset. This aligns with the findings in[51], as the data splits were designed to ensure that tissue types have similar distributions across splits, and we adhered to the same splitting methodology. For the BRCA-M2C dataset, label shifts are observed across all subsets. As shown in Table 9, we did not observe label shift for non-cell graph datasets such as coauthor networks. In the Coauthorship networks, the percentage of features with covariate shift was nearly 0, indicating minimal distributional differences between the training and test datasets. The absence of substantial covariate

| Dataset | Subset | Mean KS statistic | Chi-squared | p-value |
|---|---|---|---|---|
| TB | Train-Val | 0.0295 | 0.0085 | 0.9262 |
| | Train-Test | 0.0588 | 3555.013 | 0 |
| | Val-Test | 0.0532 | 1480.303 | 0 |
| Placenta | Train-Val | 0.2950 | 0.1271 | 1 |
| | Train-Test | 0.3160 | 0.1419 | 1 |
| | Val-Test | 0.0330 | 0.0629 | 1 |
| BRCA-M2C | Train-Val | 0 | 0.00010485247150756805 | 0.999 |
| | Train-Test | 0.0620 | 1035.9097697143552 | 1.1351457610072601e-225 |
| | Val-Test | 0.0741 | 341.3532652686813 | 7.517618058244497e-75 |

**Table 8**. Comparison of mean KS statistic, chi-squared, and p-value across datasets and subsets.

| Dataset | Subset | Chi-squared | p-value |
|---|---|---|---|
| CoAuthorPhysics | Train-Val | 3.91529 | 0.4175 |
| | Train-Test | 3.91878 | 0.4171 |
| | Val-Test | 4.87626 | 0.3002 |
| CoAuthorCS | Train-Val | 12.0746822 | 0.60030517 |
| | Train-Test | 12.309576 | 0.581456575 |
| | Val-Test | 11.29236684 | 0.662930907 |
| Synthetic Dataset | Train-Val | 3.273938 | 0.19456 |
| | Train-Test | 0.71801 | 0.6983 |
| | Val-Test | 0.5464 | 0.76091 |

**Table 9**. Comparison of Chi-squared and p-value across non-cell-based datasets and subsets.

| Model | Acc_Train ± std | Acc_Val ± std | Acc_test ± std | F1_train ± std | F1_Val ± std | F1_test ± std |
|---|---|---|---|---|---|---|
| Teacher | 0.974449 ± 0.002941 | 0.963229 ± 0.002287 | 0.902489 ± 0.001697 | 0.974466 ± 0.002947 | 0.963268 ± 0.002289 | 0.902543 ± 0.001731 |
| ExtraTrees trained on hard labels | 0.9252±0.0010 | 0.919661 ±0.0008 | 0.78278 ± 0.0017 | 0.92474 ±0.00102 | 0.919 ±0.00081 | 0.78084±0.0016 |
| **ExtraTrees trained on logits** | **0.9179± 0.00066** | **0.9134±0.00111** | **0.7873±0.00316** | **0.9173±0.00066** | **0.91272±0.001132** | **0.78531±0.00324** |
| XGBoost trained on hard labels | 0.947±0.0000 | 0.931±0.0000 | 0.793±0.0000 | 0.9469±0.0000 | 0.9307±0.0000 | 0.7922±0.0000 |
| **XGBoost trained on logits** | **0.943±0.0000** | **0.930±0.0000** | **0.806±0.0000** | **0.9423±0.0000** | **0.9298±0.0000** | **0.8049±0.0000** |
| Random Forest trained on hard labels | 0.9431± 0.00098 | 0.931±0.0014 | 0.7770± 0.00056 | 0.9428±0.0009 | 0.9306 ± 0.0014 | 0.7753 ± 0.00064 |
| **Random Forest trained on logits** | **0.933± 0.00233** | **0.922±0.002** | **0.7942±0.0063** | **0.9327±0.0023** | **0.9217±0.00285** | **0.7927±0.00680** |
| HistGrad trained on hard labels | 0.964±0.0008 | 0.947±0.0009 | 0.788±0.0028 | 0.9637±0.0008 | 0.9467±0.0009 | 0.7864±0.0028 |
| **HistGrad trained on logits** | **0.948±0.0002** | **0.938±0.0004** | **0.807±0.0046** | **0.9478±0.0002** | **0.9373±0.0004** | **0.8055±0.0063** |
| LightGBM trained on hard labels | 0.962±0.0000 | 0.944±0.0000 | 0.786±0.0000 | 0.9616±0.0000 | 0.9438±0.0000 | 0.7847±0.0000 |
| **LightGBM trained on logits** | **0.949±0.0003** | **0.936±0.0006** | **0.813±0.0016** | **0.9488±0.0004** | **0.9357±0.0006** | **0.8121±0.0014** |

**Table 10**. Model performance-TB dataset. *Note: Values in bold denote the performance of student models that learned well from the teacher model and outperformed their counterparts trained on hard labels. Std denotes the standard deviation*

shift in the coauthor networks was further supported by the performance of GNNs, where the test performance did not show a significant drop compared to the training performance. In contrast, a significant covariate shift was observed in the synthetic dataset generated by us, where 100% of the test features demonstrated a shift due to the Gaussian noise we introduced.

## Performance of student models trained on TB dataset

For this dataset, the maximum depth for HistGradientBooster, XGBoost, Random Forest, and LightGBM was set to 12. For ExtraTrees, it was set to 16. As the dataset was complex, we set the maximum number of leaf nodes to 300. Table 10 represents the performance results of various models on the training, validation, and test datasets. We do see a drop in performance on the test set. This drop is attributed to covariate and label shift, as explained in detail under "Covariate and label shift across datasets". Based on the comprehensive evaluation of various models, LightGBM achieved the best performance as a student model. HistGradientBooster emerged as the next best-performing model. Figure 5 displays the plot of the performance metrics for various models. We did not apply post-hoc calibration techniques, such as temperature scaling or isotonic regression, because the

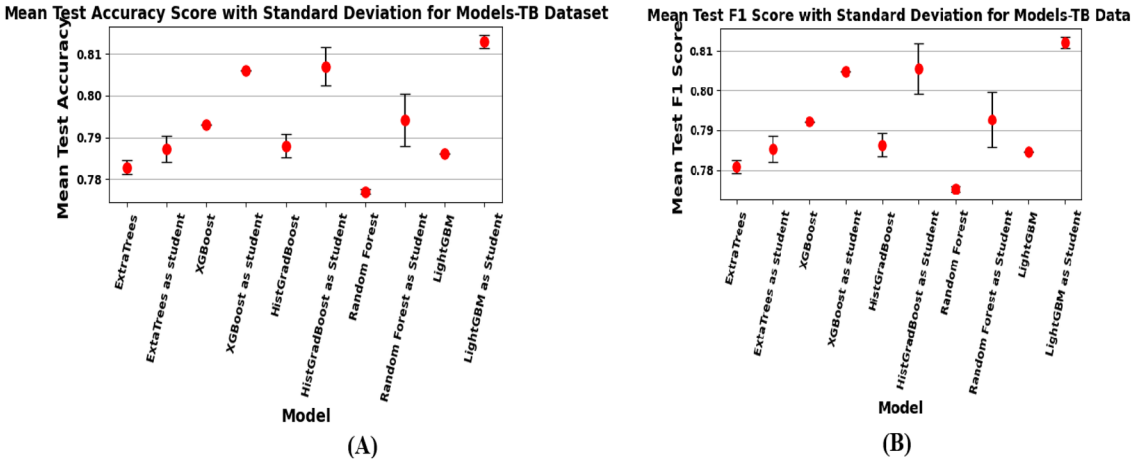


**Figure 5**. Performance of best performing student models and their counterparts on the test set-TB. We see student models outperforming their counterparts.

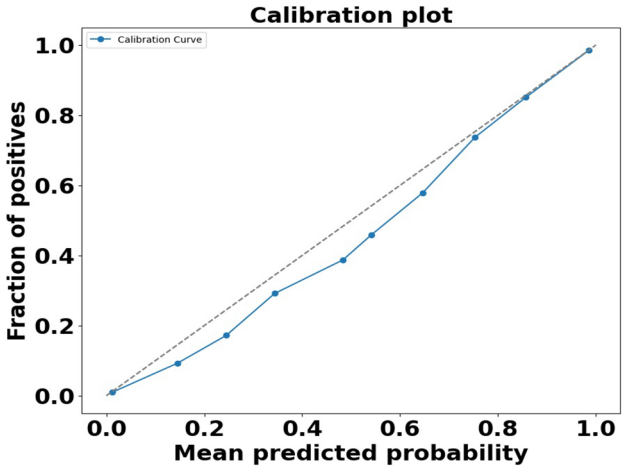


**Figure 4**. Calibration plot of raw logits converted to probabilities for positive class-TB dataset.

| Model | Best F1 score | Number_of_parameters | Distillation quality score | % Inc/Dec/NC |
|---|---|---|---|---|
| Random Forest as Student | 0.79950 | 1945.89 | 0.0669 | 3.04↑ |
| LightGBM as Student | 0.8135 | 7063.295 | 0.0829 | 3.67 ↑ |
| HistGradientBooster as Student | 0.8118 | 7166.25 | 0.084355 | 2.863↑ |
| ExtraTrees as student | 0.78855 | 1143.813 | 0.0693 | 0.78↑ |
| XGBoost as student | 0.8049 | 4399.565 | 0.0753 | 1.603 ↑ |

**Table 11**. Distillation quality scores of various student models (TB) and analysis of performance variations (F1 Score): percentage increase, decrease, or no change in student models relative to their counterparts trained on hard labels.

probabilities obtained from logits were reasonably well-calibrated. This is evident in Fig. 4, where the calibration curve is close to the diagonal. Moreover, in binary classification, the relationship between the predicted probability and the actual probability of the positive class is inherently more straightforward than in multi-class classification. Table 11 presents the distillation quality scores. The table shows that all student models exhibited performance gains, with each demonstrating a higher test F1 score than its counterpart trained on hard labels. Using teacher logits improves student performance by capturing important graph context. However, since the teacher directly leverages neighbor aggregation in the high homophily setting, students relying solely on node features may not fully match its performance. This is why we refer to these student models as partial proxies for

the teacher. While the teacher-model architectures remain consistent with our prior work[50], this paper (Table 10) presents an evaluation of these baselines on a slightly different dataset. Notably, all performance metrics are now computed globally across the entire dataset, unlike the batch-level average accuracy scores reported in[50].

## Performance of student models trained on placenta dataset

The dataset exhibited extreme class imbalance, so we employed a weighted cross-entropy loss while training the teacher model. The class weights were determined based on the recommendations provided in the paper[51]. These weights were applied to ensure fair treatment of minority classes during training. When training the student models using the logits from the teacher, we did not explicitly use these weights, as the logits already encapsulated the class imbalance information. However, we applied the same weights to maintain consistency and address the class imbalance to train the counterpart models that used hard labels.

For this dataset, the maximum depth for HistGradientBooster, Random Forest, and XGBoost was set to 12. For LightGBM and ExtraTrees, it was set to 16. Additionally, the maximum number of leaves was fixed at 50 for all models. The model performances are summarized in Table 12. As shown in their paper[51], all scalable GNN architectures–GraphSAGE, ClusterGCN, GraphSAINT, ShaDow, and SIGN, performed within 2% mean accuracy of each other, with none surpassing 65% accuracy. This indicates that the challenges observed are not unique to our approach but are inherent to the highly imbalanced and complex nature of the dataset.

The calibration plots of logits for the teacher model, without using and using weighted cross-entropy loss, are shown in Fig. 6. We notice that the weighted cross-entropy rebalanced the teacher model's focus. It improved the calibration for minority classes (Class 3 and Class 8) while causing a decrease in calibration for well-represented classes. Without the weighted cross-entropy loss, the teacher tended to favor majority classes, assigning more reliable probabilities while struggling to calibrate probabilities for the minority classes. Since our primary objective was to enhance generalization and ensure equal importance for all classes(critical for accurately representing placenta function), we employed the weighted cross-entropy loss during the teacher training. The calibration curves obtained after using weighted cross-entropy loss are shown in Fig. 7.

| Model | Acc_Train ± std | Acc_Val ± std | Acc_test ± std | F1_train ± std | F1_Val ± std | textbfF1_test ± std |
|---|---|---|---|---|---|---|
| Teacher | 0.5083 ± 0.0090 | 0.5087± 0.0116 | 0.5463 ±0.0114 | 0.4939 ±0.0118 | 0.4662 ±0.0110 | 0.5043 ±0.0066 |
| ExtraTrees trained on hard labels | 0.3768 ± 0.00125 | 0.4188± 0.0010 | 0.4706 ± 0.00167 | 0.31061 ± 0.00286 | 0.3367 ± 0.0012 | 0.3905 ± 0.0016 |
| **ExtraTrees trained on logits** | **0.4311 ± 0.0002** | **0.4287 ± 0.0005** | **0.4595 ± 0.0024** | **0.4152 ± 0.0010** | **0.4070 ± 0.0017** | **0.4458 ± 0.0004** |
| ExtraTrees trained on Calibrated probs using IR | 0.4599 ± 0.0002 | 0.4554 ± 0.0004 | 0.5074 ± 0.0001 | 0.3788 ± 0.0005 | 0.3547 ± 0.0010 | 0.4151 ± 0.0006 |
| ExtraTrees trained on Calibrated probs using temp scaling-BS | 0.4331 ± 0.0004 | 0.4505 ± 0.0005 | 0.4955 ± 0.0003 | 0.3722 ± 0.0009 | 0.3674 ± 0.0013 | 0.4162 ± 0.0018 |
| XGBoost trained on hard labels | 0.4083 ± 0.0000 | 0.4238 ± 0.0000 | 0.4703 ± 0.0000 | 0.3736 ± 0.0000 | 0.3695 ± 0.0000 | 0.4203 ± 0.0000 |
| **XGBoost trained on logits** | **0.4253 ± 0.0000** | **0.4227 ± 0.0000** | **0.4457 ± 0.0000** | **0.4179 ± 0.0000** | **0.4099 ± 0.0000** | **0.4427 ± 0.0000** |
| XGBoost trained on Calibrated probs using IR | 0.4652 ± 0.0000 | 0.4590 ± 0.0000 | 0.5108 ± 0.0000 | 0.3848 ± 0.0000 | 0.3578 ± 0.0000 | 0.4169 ± 0.0000 |
| XGBoost trained on calibrated probs using temp scaling-BS | 0.4384 ± 0.0000 | 0.4536 ± 0.0000 | 0.4985 ± 0.0000 | 0.3784 ± 0.0000 | 0.3692 ± 0.0000 | 0.4182 ± 0.0000 |
| HistGrad trained on hard labels | 0.4227 ± 0.0011 | 0.4209 ± 0.0010 | 0.4621 ± 0.0005 | 0.4029 ± 0.0013 | 0.3838 ± 0.0014 | 0.4286 ± 0.0012 |
| **HistGrad trained on logits** | **0.4277 ± 0.0007** | **0.4255 ± 0.0008** | **0.4522 ± 0.0005** | **0.4198 ± 0.0007** | **0.4128 ± 0.0008** | **0.4482 ± 0.0006** |
| HistGrad trained on Calibrated probs using IR | 0.4671 ± 0.0002 | 0.4598 ± 0.0003 | 0.5106 ± 0.0004 | 0.3891 ± 0.0004 | 0.3605 ± 0.0005 | 0.4186 ± 0.0006 |
| HistGrad trained on calibrated probs using temp scaling-BS | 0.4388 ± 0.0004 | 0.4537 ± 0.0002 | 0.4981 ± 0.0005 | 0.3838 ± 0.0004 | 0.3737 ± 0.0010 | 0.4216 ± 0.0014 |
| Random Forest trained on hard labels | 0.38124 ± 0.0001 | 0.4222 ± 0.00124 | 0.4711 ± 0.0011 | 0.3203 ± 0.00129 | 0.34376 ± 0.0012 | 0.39527 ± 0.0004 |
| **Random Forest trained on logits** | **0.4296 ± 0.0015** | **0.4274 ± 0.0012** | **0.4620 ± 0.0049** | **0.4106 ± 0.0002** | **0.4016 ± 0.0011** | **0.4444 ± 0.0033** |
| Random Forest trained on Calibrated probs using IR | 0.4571 ± 0.0007 | 0.4546 ± 0.0006 | 0.5065 ± 0.0007 | 0.3714 ± 0.0007 | 0.3497 ± 0.0005 | 0.4097 ± 0.0008 |
| Random Forest trained on Calibrated probs using temp scaling-BS | 0.4348 ± 0.0011 | 0.4505 ± 0.0004 | 0.4984 ± 0.0007 | 0.3613 ± 0.0012 | 0.3556 ± 0.0013 | 0.4081 ± 0.0018 |
| LightGBM trained on hard labels | 0.4242 ± 0.0008 | 0.4241 ± 0.0021 | 0.4667 ± 0.0009 | 0.4004 ± 0.0010 | 0.3814 ± 0.0027 | 0.4276 ± 0.0012 |
| **LightGBM trained on logits** | **0.4278 ± 0.0008** | **0.4250 ± 0.0007** | **0.4503 ± 0.0015** | **0.4198 ± 0.0006** | **0.4118 ± 0.0006** | **0.4460 ± 0.0012** |
| LightGBM trained on Calibrated probs using IR | 0.4670 ± 0.0002 | 0.4596 ± 0.0002 | 0.5105 ± 0.0002 | 0.3896 ± 0.0006 | 0.3606 ± 0.0010 | 0.4191 ± 0.0009 |
| LightGBM trained on Calibrated probs using temp scaling-BS | 0.4390 ± 0.0001 | 0.4536 ± 0.0002 | 0.4979 ± 0.0002 | 0.3842 ± 0.0001 | 0.3736 ± 0.0007 | 0.4213 ± 0.0004 |

**Table 12**. Model performance-placenta dataset. Note: The logits represent the raw outputs of the teacher model. IR denotes Isotonic Regression, BS denotes Brier score reduction, and LL denotes log loss reduction. Values in bold denote the performance of student models that learned well from the teacher model and outperformed their counterparts trained on hard labels. Std denotes the standard deviation. Different class-weighting than the one applied here may yield different teacher logits and, consequently, different student-model performances

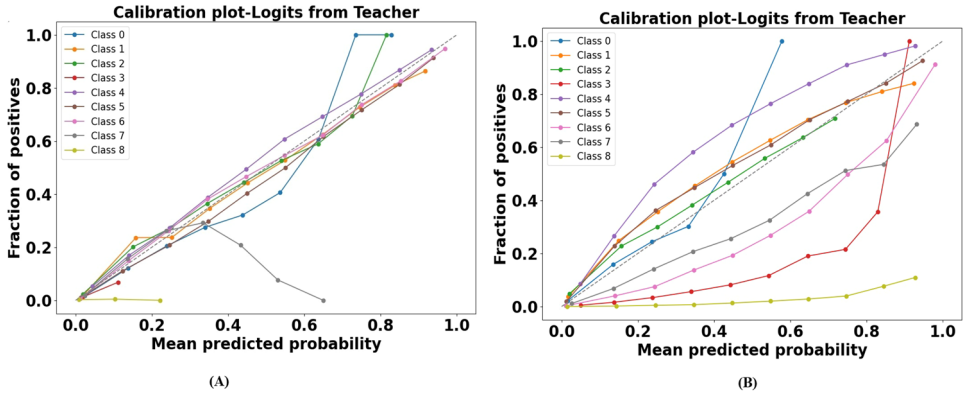


**Figure 6**. (**A**) Calibration plot: probabilities derived from raw logits of the teacher model trained with standard cross-entropy loss. (**B**) Calibration plot: probabilities derived from raw logits of the teacher model trained with weighted cross-entropy loss.

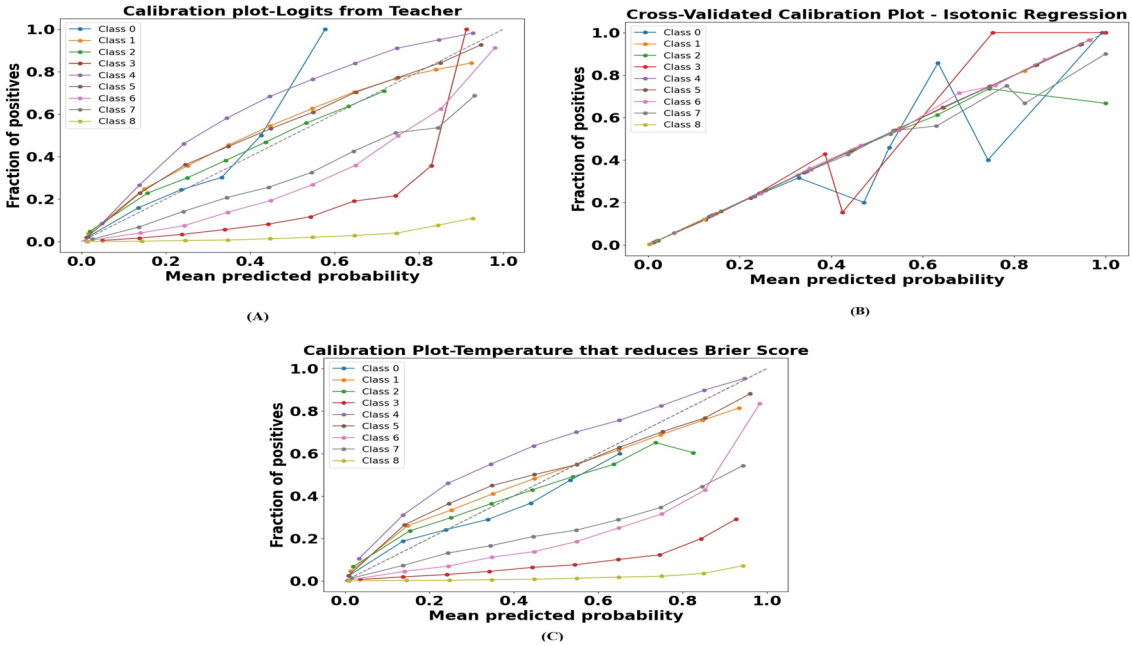


**Figure 7**. (**A**) Calibration plot: raw logits converted to probabilities. (**B**) Calibration plot after applying isotonic regression. (**C**) Calibration plot after applying temperature scaling with a temperature that reduces Stratified Brier score.

As highlighted in the paper[121], the effect of temperature scaling in the presence of class imbalance has not been adequately explored. Our experiments found that using a temperature that minimized the log loss was not suitable. Instead, we relied on the temperature that minimized the stratified Brier score. Additionally, we observed that isotonic regression behaved unstably under extreme class imbalance. Specifically, Classes 3 and 8, being minority classes, exhibited disproportionately high scores. This observation aligns with the findings of[122], where the authors noted that isotonic regression tends to perform unstably in highly imbalanced scenarios. We also found that calibration achieved using the temperature that minimized the stratified Brier score was superior to isotonic regression. This improvement was reflected in the performance of student models, as the temperature-scaled probabilities provided better guidance than the probabilities obtained from isotonic regression. The performance achieved with uncalibrated logits was higher than that obtained with calibrated logits after post-hoc calibration. We attribute this to the insufficient amount of data available per class, which is critical for the effectiveness of these calibration methods. This observation aligns with the findings of[107], where the authors noted that post-hoc calibration methods require sufficient data per class to perform effectively. The stratified Brier scores are reported in Table 13. The temperature obtained through our temperature scaling process resulted in a worse stratified Brier score for the minority class 3. This highlights the limitation of standard temperature scaling in addressing class-specific miscalibration. We recommend an advanced temperature scaling approach

| Method/Data | Stratified Brier Score |
|---|---|
| Before Calibration | Class 0: 0.4236 |
| | Class 1: 0.2080 |
| | Class 2: 0.2875 |
| | Class 3: 0.3420 |
| | Class 4: 0.3004 |
| | Class 5: 0.1902 |
| | Class 6: 0.0985 |
| | Class 7: 0.3315 |
| | Class 8: 0.1338 |
| | **Overall: 0.2573** |
| Isotonic Regression | Class 0: 0.4117 |
| | Class 1: 0.1817 |
| | Class 2: 0.2634 |
| | Class 3: 0.4695 |
| | Class 4: 0.2294 |
| | Class 5: 0.1612 |
| | Class 6: 0.1477 |
| | Class 7: 0.3938 |
| | Class 8: 0.4662 |
| | **Overall: 0.3027** |
| Temp Scaling-Reduces Brier Score | Class 0: 0.4410 |
| | Class 1: 0.1911 |
| | Class 2: 0.2799 |
| | Class 3: 0.3485 |
| | Class 4: 0.3038 |
| | Class 5: 0.1841 |
| | Class 6: 0.0921 |
| | Class 7: 0.3246 |
| | Class 8: 0.1094 |
| | **Overall: 0.2527** |

**Table 13**. Stratified brier scores-placenta dataset.

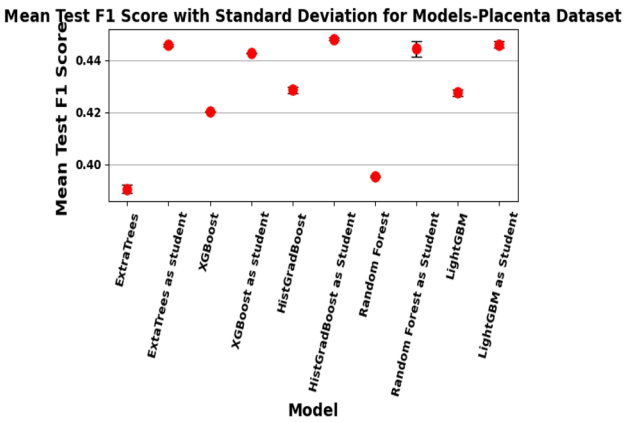


**Figure 8**. Performance of best performing student models and their counterparts on the test set-placenta. We see student models outperforming their counterparts.

designed to improve class-wise calibration which is necessary to address this issue effectively[123]. Contrary to expectations, using calibrated logits did not improve training set performance (eventhough the logits were specifically calibrated for this training set). We hypothesize that temperature scaling likely compressed the logits to an extent that it masked the tree model's optimal decision splits. Given the dataset's imbalance, the weighted F1 score is a more reliable metric to evaluate model performance. As observed from the plots, student models trained using teacher logits consistently outperform their counterparts trained on hard labels. Figure 8 presents

a comparative analysis of the performance of the best-performing student models and their counterparts on the test set. Standard deviations are represented by the error bars.

Table 14 highlights the trade-off between model complexity, performance, and distillation quality. Models with fewer parameters, such as Random Forest and ExtraTrees, are simpler. ExtraTrees has the lowest parameter count. In contrast, LightGBM and XGBoost achieve the best F1 scores, indicating superior predictive performance. The distillation quality score balances model complexity and performance. LightGBM, HistGradientBooster, and XGBoost perform well. However, their higher complexity results in slightly worse distillation scores. As indicated in Table 14, all models benefit from knowledge distillation, consistent with the trend observed in our TB dataset. Among the student models, the Random Forest and ExtraTrees regressors benefited the most, while HistGradientBooster emerged as the best-performing model overall.

## Performance of student models trained on TCGA breast cancer cell classification dataset

For this dataset, the maximum depth for HistGradientBooster, Random Forest, and XGBoost was set to 12. For LightGBM and ExtraTrees, it was set to 16. Additionally, the maximum number of leaves was fixed at 50 for all models. We observe that most student models outperform the teacher. We primarily attribute this trend to the relatively smaller training data or the low homophily. Despite this limitation, the teacher's logits remain meaningful in guiding the smaller student models. The smaller students benefit from a two-fold advantage: their reduced size allows for simplicity, while they leverage the guidance of the large teacher to achieve superior performance. The performance of the models is tabulated in the Table 15. We observed that raw logits consistently outperformed calibrated probabilities for most models. We attribute it to the fact that it preserved a good balance between resolution and reliability when compared to calibrated probabilities obtained from isotonic regression, which exhibited higher reliability but a lower resolution. The calibration plots are shown in the Fig. 9. The stratified Brier scores and log loss values achieved are tabulated in the Table 16. When calibrated probabilities from temperature scaling were used, we observed a drop in student model performance on the test set. This could be because, although temperature scaling improved the calibration of teacher logits on the validation folds during 5-fold cross-validation, the resulting calibration might not have generalized well to the test set under distribution shift[117]. The distillation quality scores and the effectiveness of the logits from teacher models in enhancing or limiting the student's classification capabilities are presented in Table 17. Even though the students outperform their teacher, we do not observe a perfect zero distillation score. This is because our evaluation assigns equal importance to performance and complexity. Since the students retain some level of complexity, the score is not entirely zero but remains very close to zero. Figure 10 presents a comparative analysis of the performance of the best-performing student models and their counterparts on the test set.

## Ablation study results

*Feature importance: comparing students trained with and without teacher guidance*

We performed this ablation study on the TB and BRAC_M2C datasets, as these were the datasets from which we extracted local cell graphs and morphological features. In our experiments with the TB dataset, the student model guided by the teacher placed greater emphasis on morphological characteristics than its counterpart guided by hard labels. This can be seen in the Figs. 11 and 12. Interestingly, the teacher-guided student prioritized features such as contrast, area, mean_image, circularity, and homogeneity, along with local cell graph features, which align with real-life considerations. For example, pathologists often use circularity to distinguish AFBs from the nucleus of activated macrophages. AFBs are rod-shaped and less circular compared to the nucleus of macrophages. This is also seen in the SHAP plots. We also notice higher contrast values for AFB. AFBs demonstrate distinct transitions or boundaries between texture regions. This likely stems from its unique cell wall properties, creating sharp intensity changes and well-defined structures. As per the expert, the staining procedure, which uses a red dye for AFB and a blue dye for surrounding tissue, may further contribute to the higher gray-level co-occurrence matrix (GLCM) contrast observed for AFB. Eccentricity is the maximum distance of a node from all other nodes in a graph. For AFB, higher eccentricity reflects their spatial isolation within tissue networks. It aligns with their biological behavior of immune evasion and persistence in host tissues. In contrast, the model trained on hard labels emphasized features like node clustering, hub-promoted index, and eccentricity. AFBs exhibit higher node clustering coefficients because they tend to form local clusters or communities. The AFBs also have a higher hub-promoted index. These nodes are pivotal in connecting various parts of the network, acting as a hub. According to domain experts, this aligns with the biological context, where the presence of AFB triggers the host's inflammatory responses and activates the immune system. For the BRAC_M2C dataset, the

| Model | Best F1 Score | Number_of_parameters | Distillation quality score | % Inc/Dec/NC |
|---|---|---|---|---|
| Random forest as student | 0.4477 | 1333.0885 | 0.07528 | 13.1 ↑ |
| LightGBM as student | 0.4472 | 2654.90 | 0.089 | 4.2910 ↑ |
| HistGradientBooster as Student | 0.4488 | 2532.403 | 0.0862 | 4.42066 ↑ |
| ExtraTrees as student | 0.4462 | 309.60115 | 0.0664 | 13.8↑ |
| XGBoost as student | 0.4427 | 2762.976 | 0.09459 | 5.329↑ |

**Table 14**. Distillation quality scores of various student models (placenta) and analysis of performance variations (F1 score): percentage increase, decrease, or no change in student models relative to their counterparts trained on hard labels.

| Model | Acc_Train ± std | Acc_Val ± std | Acc_test ± std | F1_train ± std | F1_Val ± std | F1_test ± std |
|---|---|---|---|---|---|---|
| Teacher Model | 0.9818 ± 0.0021 | 0.9466 ± 0.0013 | 0.9111 ± 0.0138 | 0.9818 ± 0.0021 | 0.9460 ± 0.0013 | 0.9059 ± 0.0159 |
| ExtraTrees trained on hard labels | 0.9448 ±0.0020 | 0.9447 ±0.001 | 0.90392 ± 0.007 | 0.9449 ± 0.0019 | 0.94487 ± 0.0011 | 0.90752 ±0.006 |
| ExtraTrees trained on logits | 0.95027 ± 0.0031 | 0.9438 ± 0.00533 | 0.95774064 ± 0.00344 | 0.95031 ± 0.0031 | 0.94384 ± 0.0053 | 0.95788 ± 0.00344 |
| **ExtraTrees trained on Calibrated probs using IR** | **0.9512 ± 0.0025** | **0.94827 ± 0.0061** | **0.9584 ± 0.0037** | **0.9513 ± 0.0026** | **0.9483 ± 0.0061** | **0.9587 ± 0.0037** |
| ExtraTrees trained on Calibrated probs using temp scaling-BS | 0.95000 ± 0.0030 | 0.9464 ± 0.004519 | 0.95623 ± 0.003135 | 0.9500 ± 0.0030 | 0.94645 ± 0.0045 | 0.9565 ± 0.0032 |
| ExtraTrees trained on Calibrated probs using temp scaling-LL | 0.95051 ± 0.002 | 0.94643 ± 0.0041 | 0.95751 ± 0.0023 | 0.950 ± 0.0027 | 0.9464 ± 0.00422 | 0.95777 ± 0.0024 |
| XGBoost trained on hard labels | 0.9683 ± 0 | 0.9614 ± 1.11e-16 | 0.90708 ± 0 | 0.9682 ± 0 | 0.9613 ± 1.11e-16 | 0.9105 ± 0 |
| **XGBoost trained on logits** | **0.9575 ± 0** | **0.9508 ± 0** | **0.9611 ± 0** | **0.9574 ±0** | **0.9507 ± 0** | **0.9613 ±0** |
| XGBoost trained on calibrated probs using IR | 0.96689 ± 0.0000 | 0.95959 ± 0.0000 | 0.9085 ± 0.0000 | 0.96682 ± 0.0000 | 0.9595 ± 0.0000 | 0.9119 ± 0.0000 |
| XGBoost trained on calibrated probs using temp scaling-BS | 0.9653 ± 0 | 0.9573 ± 0 | 0.9073 ± 0 | 0.9653 ± 0 | 0.9572 ± 0 | 0.9109 ±0 |
| XGBoost trained on calibrated probs using temp scaling-LL | 0.9653 ± 0 | 0.9591 ±0 | 0.9084 ±0 | 0.9652 ± 0 | 0.959 ± 1.110e-16 | 0.9119 ± 1.1105e-16 |
| HistGrad trained on hard labels | 0.9843 ± 0.0002 | 0.9696 ± 0.0016 | 0.9102 ± 0.0007 | 0.984 ± 0.00024 | 0.9696 ± 0.00164 | 0.91388 ± 0.0007 |
| **HistGrad trained on logits** | **0.9648 ± 0.00115** | **0.9579 ±0.0008** | **0.9568 ±0.00064** | **0.9648 ± 0.00114** | **0.9579 ± 0.00087** | **0.9573 ± 0.00062** |
| HistGrad trained on calibrated probs using IR | 0.9743 ± 0.0001 | 0.9666 ± 0.0005 | 0.9102 ± 0.0004 | 0.9743 ± 0.0001 | 0.9666 ± 0.0005 | 0.9138 ± 0.0004 |
| HistGrad trained on calibrated probs using temp scaling-BS | 0.97440 ± 0.0002 | 0.9667 ± 0.0009 | 0.91006 ± 0.0007 | 0.9743 ± 0.0002 | 0.9667 ± 0.0009 | 0.9137± 0.0007 |
| HistGrad trained on calibrated probs using temp scaling-LL | 0.9738 ± 0.0001 | 0.966 ± 0.002 | 0.9107 ± 0.0006 | 0.9738 ± 0.0001 | 0.966 ± 0.0025 | 0.9142 ± 0.00056 |
| Random Forest trained on hard labels | 0.9537±0.0015 | 0.9469±0.0018 | 0.8999±0.00168 | 0.9615±0.001267 | 0.953739±0.0015 | 0.904±0.0015 |
| **Random Forest trained on logits** | **0.9557±0.0007** | **0.9474±0.0016** | **0.9229±0.0063** | **0.9558±0.0007** | **0.9474±0.0016** | **0.9254±0.0059** |
| Random Forest trained on calibrated probs using IR | 0.9688 ± 0.0005 | 0.9597 ± 0.0002 | 0.9069 ± 0.0013 | 0.9688 ± 0.0005 | 0.9597 ± 0.0001 | 0.9107 ± 0.0012 |
| Random trained on calibrated probs using temp scaling-BS | 0.9688±0.0004 | 0.9597±0.0008 | 0.9070±0.0012 | 0.9687±0.0004 | 0.9597±0.0008 | 0.9108±0.0011 |
| Random trained on calibrated probs using temp scaling-LL | 0.9688±0.0008 | 0.9593±0.0013 | 0.9074±0.0006 | 0.9688±0.0008 | 0.9593±0.0013 | 0.9112±0.0005 |
| LightGBM trained on hard labels | 0.9793 ± 0.0 | 0.9683 ± 0.0 | 0.9103 ± 0.0 | 0.9793 ± 0.0 | 0.9683 ± 0.0 | 0.9139 ± 0.0 |
| **LightGBM trained on logits** | **0.9648 ± 0.0** | **0.9555 ± 0.0** | **0.9626 ± 0.0** | **0.9648 ± 0.0** | **0.9554 ± 0.0** | **0.9629 ± 0.0** |
| LightGBM trained on Calibrated probs using IR | 0.9746 ± 0.0000 | 0.9641 ± 0.0000 | 0.9101 ± 0.0000 | 0.9746 ± 0.0000 | 0.9641 ± 0.0000 | 0.9138 ± 0.0000 |
| LightGBM trained on Calibrated probs using temp scaling-BS | 0.9742 ± 0.0 | 0.9656 ± 0.0 | 0.9104 ± 0.0 | 0.9742 ± 0.0 | 0.9655 ± 0.0 | 0.9140 ± 0.0 |
| LightGBM trained on Calibrated probs using temp scaling-LL | 0.9742 ± 0.0 | 0.9656 ± 0.0 | 0.9112 ± 0.0 | 0.9742 ± 0.0 | 0.9656 ± 0.0 | 0.9148 ± 0.0 |

**Table 15**. Model performance-breast cancer dataset. Note: The logits represent the raw outputs of the teacher model. IR denotes Isotonic Regression, BS denotes Brier score reduction, LL denotes Log Loss reduction. Values in bold denote the performance of student models that learned well from the teacher model and outperformed their counterparts trained on hard labels. Std denotes the standard deviation

LightGBM model emerged as the best-performing model and was consequently used for the analysis. When trained on hard labels, the LightGBM model emphasized features such as degree, betweenness centrality, mean_all_neighbors, and Salton index. Figure 13 shows the plot of feature importance. In contrast, when guided by the teacher's logits, the model emphasized degree, and the hub promoted index. node clustering, sørensen, and eccentricity. To further evaluate the biological relevance of these features, we analyzed their contributions using SHAP plots shown in Fig. 14. Degree values were moderate for lymphocytes, lower for breast cancer cells, and highest for stromal cells because of their extensive connections. Node clustering was high for lymphocytes and breast cancer cells. This aligns with their biological behavior, as lymphocytes naturally cluster near cancer cells in immune hotspots, forming localized areas of immune activity[124]. It was lower for stromal cells as they are separated by extracellular matrix such as collagen and are not as densely clustered as lymphocytes or cancer cells. The Hub-Promoted Index (HPI) measured the overlap between neighbors of two connected nodes. It was lower for stromal cells due to their diverse and dispersed connections with fewer overlapping neighbors and higher for breast cancer cells because of their dense clustering and high number of common neighbors. The Mean_all_neighbors feature measured the average distance between a node and all its neighbors in the graph. Breast cancer cells exhibited higher values, which was likely driven by some long-distance connections with stromal cells[125]. Breast cancer cells have a higher Sørensen index when compared to lymphocytes due to their tight clustering and significant overlap of neighbors. This shows their cohesive role in the tumor microenvironment.

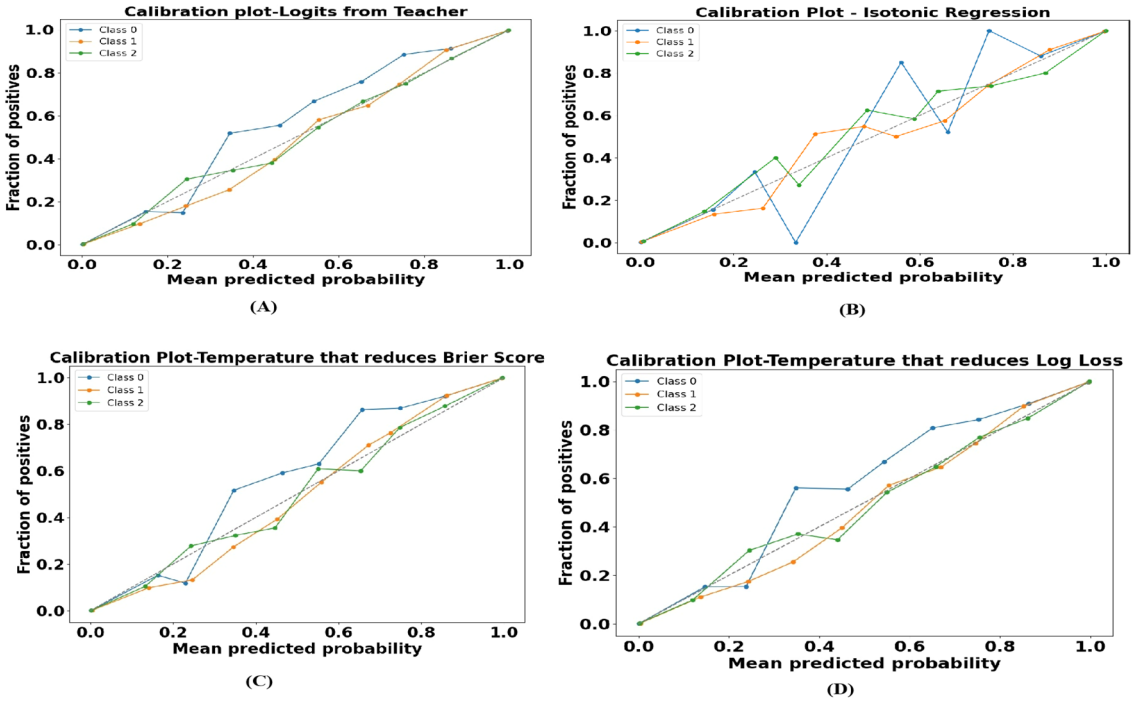


**Figure 9**. (**A**) Calibration plot: raw logits converted to probabilities. (**B**) Calibration plot after applying isotonic regression. (**C**) Calibration plot after applying temperature scaling with a temperature that reduces the Stratified Brier score. (**D**) Calibration plot after applying temperature scaling with a temperature that reduces negative log-likelihood (log loss).

| Method/Data | Stratified Brier Score | Log Loss |
|---|---|---|
| Before Calibration | Class 0: 0.01773<br>Class 1: 0.015629<br>Class 2: 0.011969<br>**Overall: 0.015111** | Class 0: 0.03479<br>Class 1: 0.05215<br>Class 2: 0.02904<br>**Overall:0.038663** |
| Isotonic Regression | Class 0: 0.016961<br>Class 1: 0.0156<br>Class 2: 0.012267<br>**Overall: 0.0149** | Class 0: 0.037611<br>Class 1: 0.055318<br>Class 2: 0.030436<br>**Overall: 0.041** |
| Temp Scaling - Reduces Brier Score | Class 0: 0.01753<br>Class 1: 0.01581<br>Class 2: 0.01187<br>**Overall: 0.0150708** | Class 0: 0.03534<br>Class 1: 0.05317<br>Class 2: 0.02959<br>**Overall: 0.039368** |
| Temp Scaling - Reduces Log Loss | Class 0: 0.01778<br>Class 1: 0.01562<br>Class 2: 0.01199<br>**Overall: 0.015127** | Class 0: 0.03474<br>Class 1: 0.05206<br>Class 2: 0.02899<br>**Overall: 0.03860** |

**Table 16**. stratified brier scores and log loss values-breast cancer dataset.

| Model | Best F1 Score | Number_of_parameters | Distillation Quality Score | % Inc/Dec/NC |
|---|---|---|---|---|
| Random Forest as Student | 0.9313 | 787.085 | 0.0047 | 2.8↑ |
| LightGBM as Student | 0.9629 | 906.29 | 0.0055 | 5.36 ↑ |
| HistGradientBooster as Student | 0.958 | 938.4595 | 0.0057 | 4.75 ↑ |
| ExtraTrees as student | 0.9624 | 269 | 0.0016 | 5.35 ↑ |
| XGBoost as student | 0.9613 | 822.25 | 0.005 | 5.58 ↑ |

**Table 17**. Distillation quality scores of various student models (breast cancer) and analysis of performance variations (F1 Score): percentage increase, decrease, or no change in student models relative to their counterparts trained on hard labels.

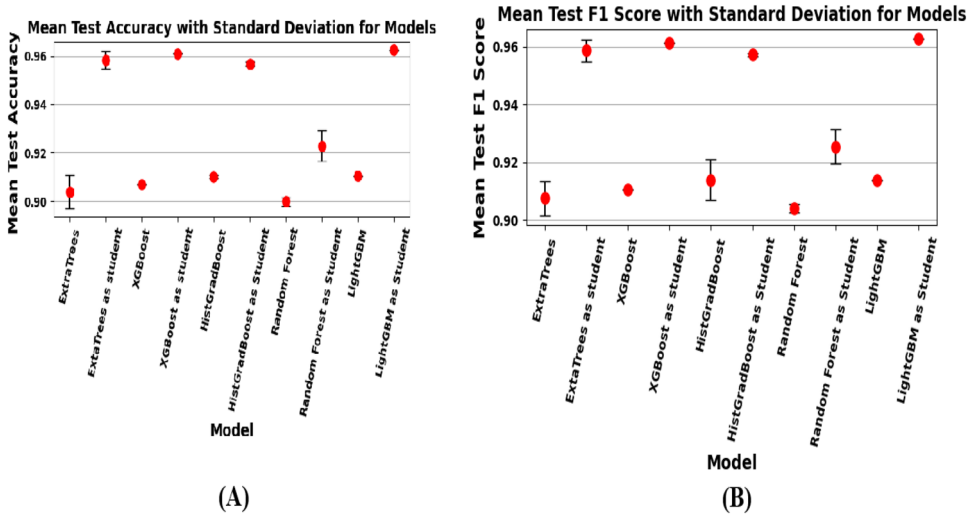


**Figure 10**. Performance of best performing student models and their counterparts on the test set-breast cancer. We see student models outperforming their counterparts.

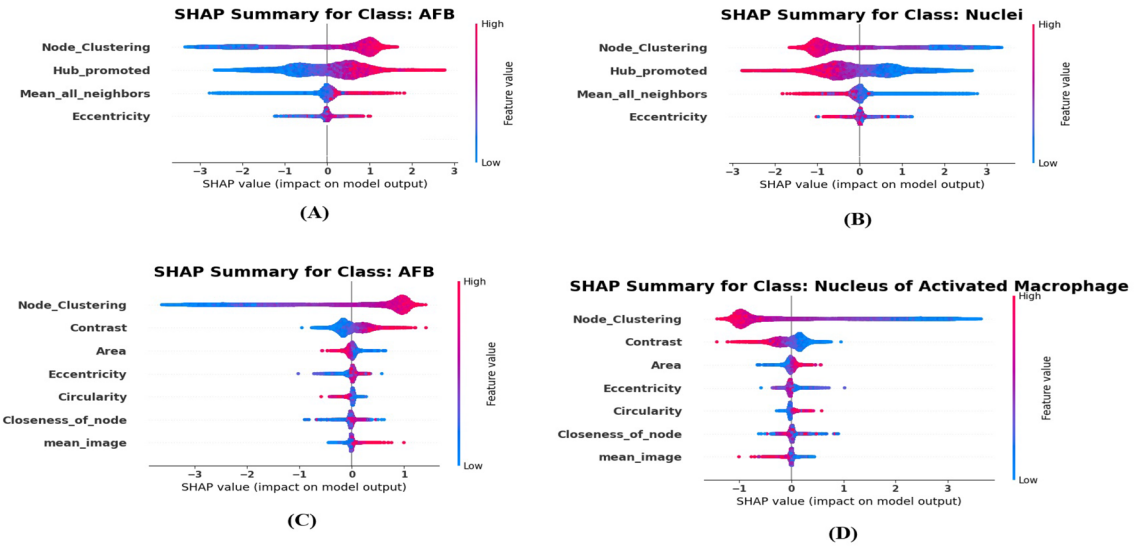


**Figure 11**. SHAP summary plots comparing feature importance for different cell types. The top row (**A**,**B**) represents features considered important when the model is trained on hard labels, while the bottom row (**C**,**D**) shows the important features when trained on logits. Note that the SHAP results do not provide sufficient evidence to clearly discern differences in the 'closeness of node' feature between AFB and nucleus of activated macrophage, limiting our ability to draw biological conclusions on this metric.

We believe the models primarily rely on features such as degree, hub-promoted index, mean_all_neighbors, node clustering, and Sørensen index to differentiate breast cancer cells from other types. In contrast, features like degree and node clustering play a key role in distinguishing stromal cells from other cell types. Observing how the model highlights betweenness centrality as a crucial feature is also interesting. As seen from the plots, this metric is particularly high for lymphocytes, suggesting this is one of the primary features the model relies on to distinguish lymphocytes from other cell types. This may be attributed to the biological role of lymphocytes, which infiltrate tumors as part of the immune response[126]. They often localize to the interface between tumor and stromal regions, where they may be blocked from entering the tumor by soluble mediators produced by the cancer cells[127]. This placement significantly enhances their betweenness centrality and reflects their role in mediating interactions between the immune system and the tumor microenvironment.

*Training with ensembled output*

In the analysis of our ensemble model performance, where logits from CG-JKNN (primary teacher) and raw scores/predictions from the best student are combined, we observe interesting trends concerning the influence

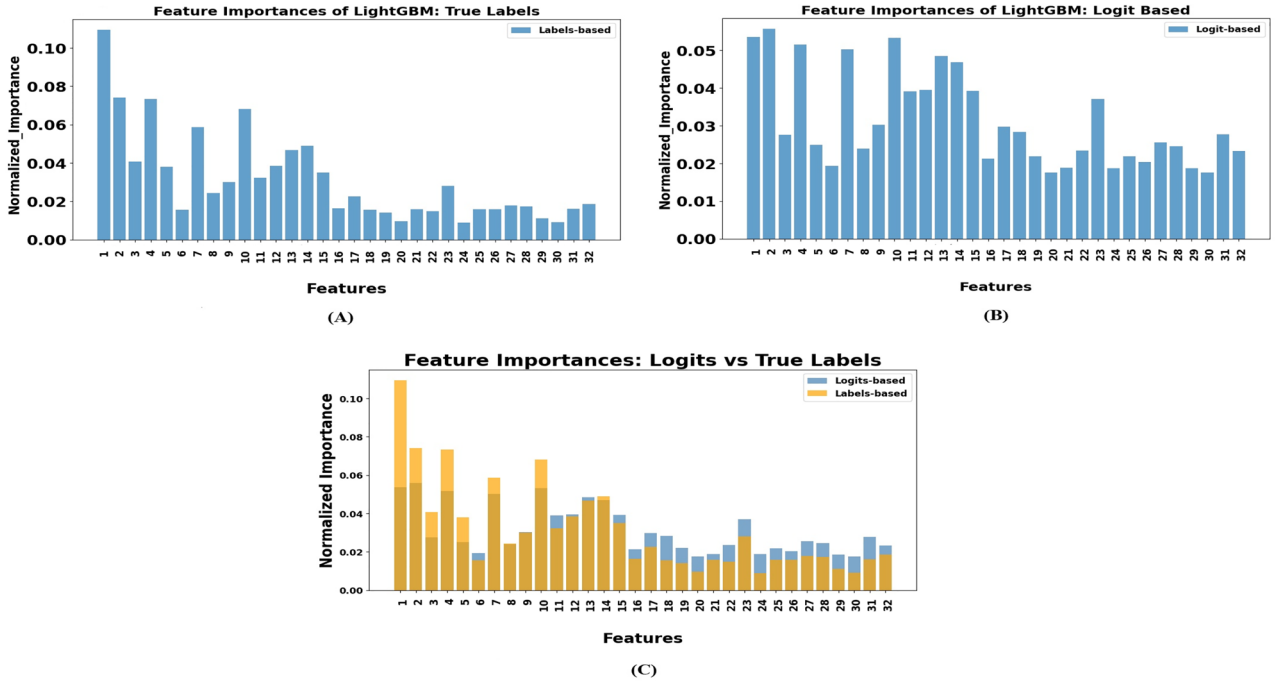


**Figure 12**. Feature importance comparison for LightGBM models trained on hard labels and logits. (**A**) Shows the feature importances when the model is trained on hard labels. (**B**) Represents the feature importances when the model is trained on logits distilled from the teacher model. (**C**) Compares the feature importances for both scenarios. The brown color indicates the overlap of feature importance between models trained on hard labels and logits. The feature numbers on the x-axis correspond to the features listed in Table 3.

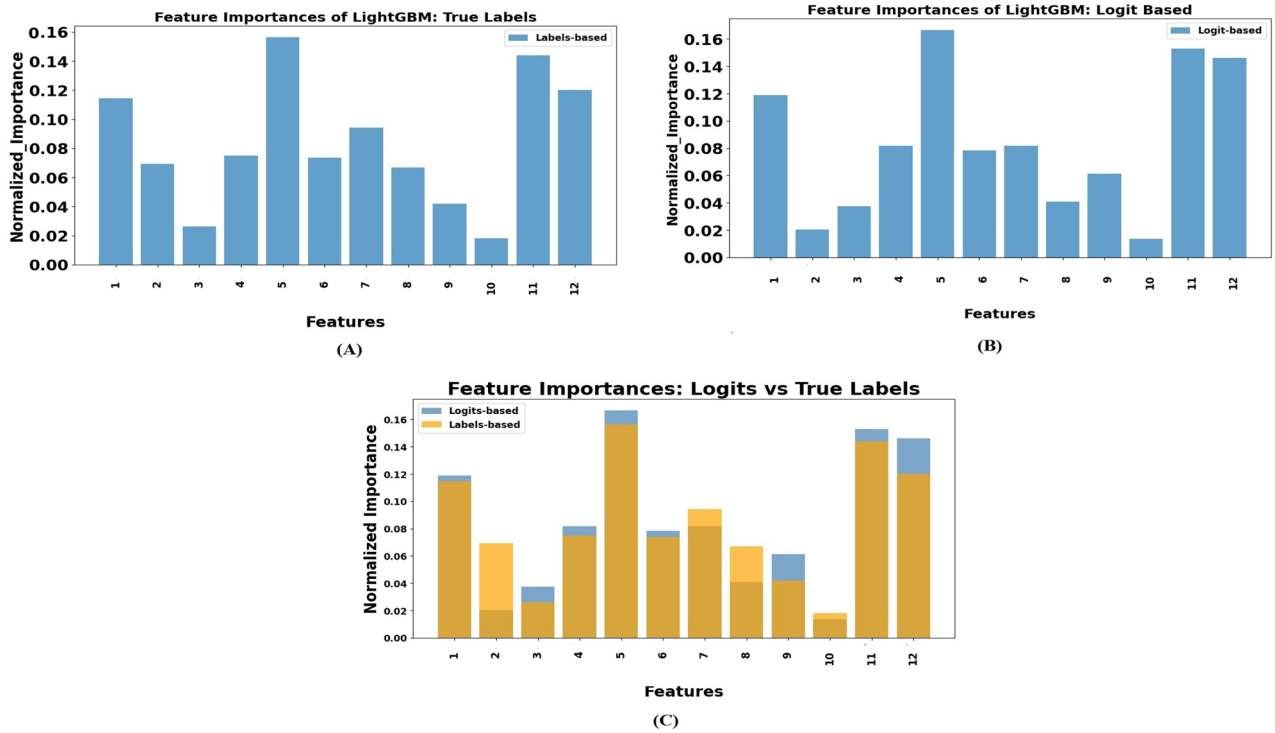


**Figure 13**. Feature importance comparison for LightGBM models trained on hard labels and logits. (**A**) Shows the feature importances when the model is trained on hard labels. (**B**) Represents the feature importances when the model is trained on logits distilled from the teacher model. (**C**) Compares the feature importances for both scenarios. The brown color indicates the overlap of feature importance between models trained on hard labels and logits.The feature numbers on the x-axis correspond to the features listed in Table 4.

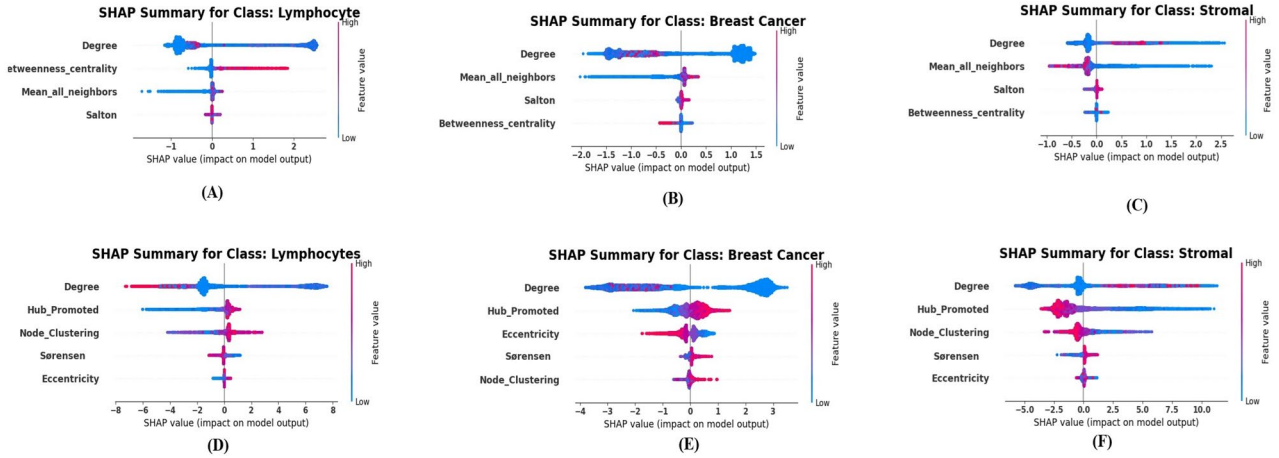


**Figure 14**. SHAP summary plots comparing feature importance for different cell types. The top row (**A**–**C**) represents features considered important when the model is trained on hard labels. The bottom row (**D**–**F**) corresponds to features considered important when the model is trained on logits.

| Dataset | Model | Weight GNN | Weight Student | Accuracy | | | F1 | | | % Inc/Dec/ NC F1 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Train | Val | Test | Train | Val | Test | |
| TB | Random Forest | 0.8 | 0.2 | 0.9022 | 0.8938 | 0.7793 | 0.9014 | 0.8926 | 0.7763 | 2.9 ↓ |
| | HistGrad | 1 | 0 | 0.9482 | 0.9384 | 0.8116 | 0.948 | 0.9377 | 0.8118 | *NC* |
| | ExtraTrees | 0.9 | 0.1 | 0.9184 | 0.9131 | 0.7912 | 0.9178 | 0.9124 | 0.7894 | 0.107↑ |
| | XGBoost | 0.9 | 0.09 | 0.9343 | 0.9235 | 0.8020 | 0.9338 | 0.9227 | 0.8006 | 0.53 ↓ |
| Placenta | LightGBM | 0.8 | 0.2 | 0.4262 | 0.4241 | 0.4485 | 0.4178 | 0.4128 | 0.4459 | ↓ 0.29 |
| | Random Forest | 0.5 | 0.5 | 0.4288 | 0.4285 | 0.4609 | 0.4087 | 0.4041 | 0.4451 | ↓ 0.58 |
| | XGBoost | 1 | 0 | 0.4253 | 0.4227 | 0.4457 | 0.4179 | 0.4099 | 0.4427 | *NC* |
| | ExtraTrees | 0.2 | 0.8 | 0.4252 | 0.4256 | 0.4562 | 0.4092 | 0.4073 | 0.4428 | ↓0.761 |
| Breast Cancer | Random Forest | 0.7 | 0.3 | 0.9543 | 0.948 | 0.9212 | 0.9544 | 0.948 | 0.9237 | ↓0.82 |
| | XGBoost | 1 | 0 | 0.9575 | 0.9508 | 0.9611 | 0.9574 | 0.9507 | 0.9613 | NC |
| | HistGrad | 0.7 | 0.3 | 0.9623 | 0.9559 | 0.9636 | 0.9623 | 0.9558 | 0.9638 | ↑0.61 |
| | ExtraTrees | 0.8 | 0.2 | 0.9525 | 0.9479 | 0.961 | 0.9525 | 0.9479 | 0.9612 | ↓0.12 |

**Table 18**. Performance of students with ensembled outputs for TB, placenta, and breast cancer datasets. The best-performing student (based on the performance and its low variability across multiple runs) for the TB dataset was LightGBM, for the Placenta dataset it was HistGrad, and for the Breast Cancer dataset it was LightGBM.

of different teachers on student models as shown in the Table 18. It is important to note that the comparisons (of F1-test scores) here are made against the baseline scenario in which only the CG-JKNN teacher model guides the students. In the case of the TB and BRAC_M2C datasets, the best-performing student model was LightGBM. On the other hand, for the Placenta dataset, the best-performing student model was HistGradientBooster.

In the TB dataset, the ExtraTrees model, when taught by LightGBM along with CG-JKNN, actually exhibits an increase in test set performance in contrast to when it is solely taught by the CG-JKNN. However, the XGBoost and Random Forest show a drop in performance. This suggests that integrating LightGBM's guidance doesn't always align with the learning patterns beneficial to all student models. In the Placenta dataset, all student models benefited from CG-JKNN rather than being taught by the joint teachers HistGradientBooster and CG-JKNN.

In the case of the BRAC_M2C dataset, we observe a performance improvement in HistGradientBooster when guided by the joint teachers LightGBM and CG-JKNN. However, XGBoost prefers to be guided solely by CG-JKNN, as it assigns zero weight to the raw scores from LightGBM. Additionally, a drop in performance is observed for the ExtraTrees and Random Forest models. Based on the above results, we should note that the influence of the 'best' student's output is not universally beneficial, as their effectiveness can vary depending on the specific characteristics of the dataset and the learning dynamics of the other models being guided.

| Dataset | Hyperparameters |
|---|---|
| TB | hidden_dim=399, lr=0.009628, alpha= 0.88 |
| Placenta | hidden_dim=78,lr=0.00204, alpha=0.131 |
| Breast Cancer | hidden_dim= 120,lr=0.00748,alpha=0.0126 |

**Table 19**. Optimized hyperparameters for the ANN student model across datasets.

| Dataset | Trained on | Train Acc ±std | Val Acc ± std | Test Acc ± std | Train F1 ± std | Val F1 ± std | Test F1 ± std |
|---|---|---|---|---|---|---|---|
| TB | Hard Labels | 0.9367 ± 0.0025 | 0.9300 ± 0.0022 | 0.8191 ± 0.0017 | 0.9365 ± 0.0025 | 0.9297 ± 0.0022 | 0.8184 ± 0.0018 |
| | Logits (alpha=0.88) | 0.9393 ± 0.0020 | 0.9332 ± 0.0015 | 0.8216 ± 0.0003 | 0.9391 ± 0.0020 | 0.9330 ± 0.0015 | 0.8210 ± 0.0002 |
| | Logits (alpha=0.5) | **0.9368 ± 0.0016** | **0.9311± 0.0024** | **0.8218 ± 0.0018** | **0.9366 ± 0.0015** | **0.9308 ± 0.0025** | **0.8211 ± 0.0019** |
| Placenta | Hard Labels | 0.3918 ± 0.0010 | 0.4276 ± 0.0009 | 0.4724 ± 0.0025 | 0.3604 ± 0.0016 | 0.3758 ± 0.0019 | 0.4232 ± 0.0045 |
| | Logits (alpha=0.131) | 0.4476 ± 0.00265 | 0.4534 ± 0.00140 | 0.4970 ± 0.00170 | 0.3993 ± 0.00271 | 0.3866 ± 0.00426 | 0.4337 ± 0.00381 |
| | Logits (alpha=0.5) | **0.4621 ± 0.0014** | **0.4587 ± 0.0009** | **0.5036 ± 0.0018** | **0.4080 ± 0.0034** | **0.3885 ± 0.0029** | **0.4387 ± 0.0030** |
| Breast Cancer | Hard Labels | 0.9492 ± 0.0011 | 0.9475 ± 0.0023 | 0.9483 ± 0.0006 | 0.9492 ± 0.0011 | 0.9474 ± 0.0023 | 0.9488 ± 0.0006 |
| | Logits (alpha=0.012) | **0.9478 ± 0.0025** | **0.9447 ±0.0046** | **0.9531 ± 0.0024** | **0.9478 ± 0.0026** | **0.9446 ± 0.0045** | **0.9535 ± 0.0023** |
| | Logits (alpha=0.5) | 0.9507 ± 0.0014 | 0.9461 ±0.0015 | 0.9537 ± 0.0011 | 0.9507 ± 0.0014 | 0.9460 ± 0.0015 | 0.9541 ± 0.0011 |

**Table 20**. Evaluation of ANN performance across multiple datasets.

| Dataset | Complexity-ANN | Estimated model complexity-Best performing non-neural model | Distillation Quality Score | % Inc/Dec/NC (F1-score) in comparison to its counterpart (trained on hard labels) | % Inc/Dec/NC (F1-score) in comparison to best non-neural student model |
|---|---|---|---|---|---|
| TB | 14366 | 7063.295 | 0.11 | 0.34% ↑ | 1.17% ↑ |
| Placenta | 5781 | 2532.403 | 0.1259 | 3.27% ↑ | 1.6% ↓ |
| Breast Cancer | 1923 | 906.29 | 0.0116 | 0.6745% ↑ | 0.74 % ↓ |

**Table 21**. Comparison of ANN student models and best-performing non-neural models: complexity, distillation quality, and relative performance. Note: The LightGBM regressor achieved the best performance on the TB dataset and the breast cancer dataset, while the HistGradientBoostingRegressor was the best performing on the Placenta dataset

*A comparative analysis of knowledge distillation in neural and non-neural student models*

Table 19 summarizes the optimized hyperparameters for the ANN student model. Table 20 shows that the ANN benefits from incorporating teacher logits into the training process. For each dataset, TB, Placenta, and Breast Cancer, the models trained with teacher logits outperform those trained solely on hard labels. Moreover, when comparing different weighting schemes for the combined loss, using an equal weight ($\alpha = 0.5$) for the cross-entropy and KL divergence losses yields the best performance compared to the $\alpha$ value tuned as a hyperparameter in most cases.

From Table 21, we can observe that although the ANN student model undergoes additional hyperparameter tuning and possesses a greater number of parameters compared to the best-performing non-neural model, it does not consistently outperform the non-neural counterparts on all datasets. In particular, for the Placenta and Breast Cancer datasets, the non-neural students achieve competitive performance, while the ANN students do not significantly improve. Only in the TB dataset does the ANN student show a slight 1.17% improvement over its non-neural counterpart, but this gain comes at the cost of additional parameters. These results validate that non-neural student models are viable and competitive alternatives, achieving comparable performance with fewer parameters and less tuning. Pure logit-regression on teacher's logits could further close the performance gap, and while increasing the ANN's capacity may boost performance, it would inflate model size and undermine the goal of a lightweight student.

## Results of generalizability on non-biological graph datasets

This section investigates whether our approach is effective when applied to non-cell graph datasets. Unlike cell graphs, where morphological and graph-specific features are typically extracted, these experiments utilize only the existing features provided in the datasets.

*Performance on CoauthorPhysicsDataset*

For this dataset, the maximum depth for HistGradientBooster, Random Forest, and XGBoost was set to 12. For LightGBM and ExtraTrees, it was set to 16. Additionally, the maximum number of leaves was fixed at 50 for all models. Figure 15 shows the calibration plots illustrating the performance of various calibration techniques across different classes. Table 22 compares the stratified Brier scores and log losses for the different methods used. As seen from the Table 23, ExtraTrees, XGBoost, and Random Forest models performed better when trained on calibrated probs from the teacher model. LightGBM and HistGradientBooster performed well when trained on hard labels because the test distribution was similar to the training distribution. These models slightly overfit the training data, which, in this case, acted as a boon rather than a bane. At the same time, the regularization effect provided by the logits did not translate into improved performance. Instead, it acted as a bane, leading to slight underperformance compared to models trained on hard labels. Also, since the student versions of these models did not outperform their counterparts, we did not record their distillation score in the Table 24. Figure 16 presents the results on the CoauthorPhysics dataset. The mean test accuracy and F1 score for various models are displayed, with error bars indicating standard deviation. LightGBM and HistGradientBooster were excluded from this comparison as the student models trained using teacher logits failed to outperform their counterparts trained on hard labels, even after calibration. Among the student models, the ExtraTrees model emerged as the best student.

The distillation quality scores, computed using Eq. 13, for the student models that consequently outperformed their counterparts trained on hard labels, are tabulated in Table 24.

*Performance on CoauthorCSDataset*

This dataset exhibited extreme class imbalance. Although this dataset did not represent a biologically critical scenario where equal importance for minority and majority classes is essential, we still applied weighted cross-entropy to address the imbalance effectively with weights set inversely proportional to each class's frequency. For this dataset, the maximum depth for HistGradientBooster, Random Forest, and XGBoost was set to 12. For LightGBM and ExtraTrees, it was set to 16. Additionally, the maximum number of leaves was fixed at 50 for all models. By selecting a hyperparameter configuration that omits explicit regularization, we create a scenario where models trained on hard labels are prone to overfitting, thereby allowing us to clearly demonstrate the efficacy of using teacher logits as an implicit regularizer. Figure 17 shows the calibration plots illustrating the performance of various calibration techniques across different classes. Table 25 compares the stratified Brier scores for the different methods used. When trained on hard labels, we observe overfitting in the HistGradientBooster and LightGBM models. However, this overfitting is reduced when the models are trained on logits, suggesting that logits provide implicit regularization and improve the models' generalization capability. We also propose that the regularization effect inherently provided by the teacher model's guidance offers more effective control over model overfitting than manually tuning explicit regularization parameters. Each model benefited from different

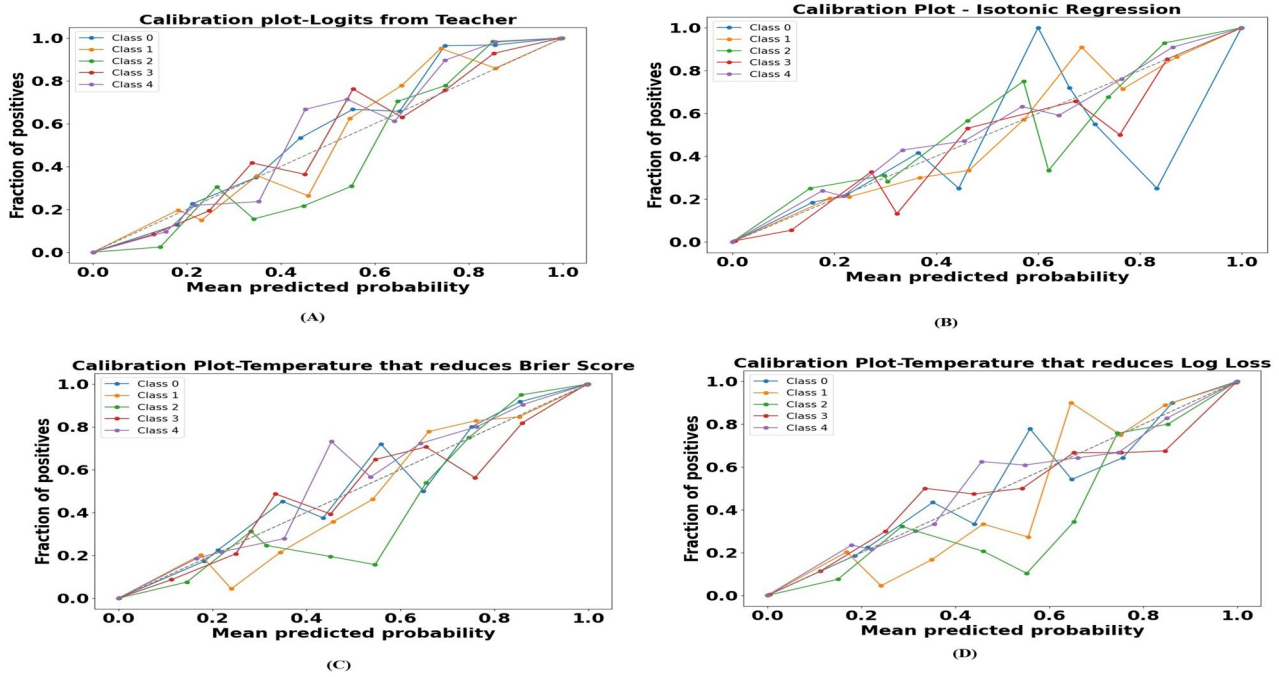


**Figure 15**. Plots along with stratified brier scores and log losses (**A**) Calibration plot: raw logits converted to probabilities. (**B**) Calibration plot after applying isotonic regression. (**C**) Calibration plot after applying temperature scaling with a temperature that reduces Stratified Brier score. (**D**) Calibration plot after applying temperature scaling with a temperature that reduces negative log-likelihood (log loss).

| Method/data | Stratified brier score | Log Loss |
|---|---|---|
| Before calibration | Class 0: 0.030995 | Class 0: 0.040620 |
| | Class 1: 0.032688 | Class 1: 0.038505 |
| | Class 2: 0.015430 | Class 2: 0.047046 |
| | Class 3: 0.036208 | Class 3: 0.028423 |
| | Class 4: 0.047892 | Class 4: 0.040356 |
| | Overall: 0.032642 | **Overall: 0.038990** |
| Isotonic regression | Class 0: 0.0300 | Class 0:0.0435 |
| | Class 1: 0.0314 | Class 1: 0.0403 |
| | Class 2: 0.0151 | Class 2: 0.0489 |
| | Class 3: 0.0370 | Class 3: 0.0296 |
| | Class 4: 0.0467 | Class 4: 0.0422 |
| | Overall: 0.03204 | **Overall: 0.0409** |
| Temp scaling-reduces brier score | Class 0: 0.0307 | Class 0: 0.0398 |
| | Class 1: 0.0326 | Class 1: 0.0383 |
| | Class 2: 0.0153 | Class 2: 0.0459 |
| | Class 3: 0.0368 | Class 3: 0.0275 |
| | Class 4: 0.0473 | Class 4: 0.0399 |
| | Overall: 0.03254 | **Overall: 0.03828** |
| Temp scaling-reduces log loss | Class 0: 0.0308 | Class 0: 0.0399 |
| | Class 1: 0.0328 | Class 1: 0.0388 |
| | Class 2: 0.0153 | Class 2: 0.0456 |
| | Class 3: 0.0377 | Class 3: 0.0276 |
| | Class 4: 0.0472 | Class 4: 0.0401 |
| | **Overall: 0.03276** | **Overall: 0.0384** |

**Table 22**. Stratified brier scores and log loss values-coauthorphysics dataset. Can you please make the overall value in bold here?. Some overall values are not bold.

calibration techniques, as shown in the Table 26. The distillation quality scores are recorded in the Table 27. Figure 18 shows the performance of the best-performing student models and their counterparts on the test set.

*Performance on synthetic dataset*

We also experimented with a synthetic dataset with three classes generated using the Barabási-Albert (BA) model[53] with a preferential attachment mechanism, which occurs in many real-world graphs[128]. Although these graphs do not fully capture the complexity of cell graphs, we utilized them due to the limited availability of cell graph-based datasets. This experiment served two primary purposes: first, to evaluate if logits provide improved guidance under distribution shift, and second, to assess the performance of post-hoc calibration methods under such shifts. The graph consisted of 60,000 nodes, with each new node attaching to five existing nodes based on the principle of linear preferential attachment. Instead of relying on random features, we computed various graph-derived features such as degree, clustering coefficient, and eigenvector centrality to capture the structural properties of the graph better. Class labels were assigned by clustering features derived from the graph using the k-means algorithm. To simulate a distribution shift, we introduced Gaussian noise to the features of the test nodes. This approach allowed us to reflect potential variations in data distribution between the training and test sets. The shift was induced synthetically to provide a controlled environment for this initial investigation, and we acknowledge that a more rigorous shifting paradigm would be a valuable next step for future studies. Our dataset had an uneven distribution of classes. However, since it wasn't a critical biological dataset, we used the standard cross-entropy loss function to train the teacher without any modifications. The algorithm is provided in 2. For this dataset, the maximum depth for HistGradientBooster, LightGBM, Random Forest, and XGBoost was set to 12. For ExtraTrees, it was set to 16. Additionally, the maximum number of leaves was fixed at 50 for all models. Table 28 shows the accuracy and F1-score for various models trained on hard labels and their student counterparts using different calibration techniques. Typically, we expect calibration to improve the guidance raw logits provide. However, in this case, we do not observe any improvement. Good calibration achieved on validation folds of the training set does not necessarily translate to good calibration on the held-out test set when a distribution shift exists[117]. This misalignment may have contributed to the observed lower performance on the test set. Our experiments revealed that when the GNN teacher was trained on extremely imbalanced data without weighted loss, its logits became biased but remained predictive for the minority class. Distilling from these raw, uncalibrated logits produced a student model with the highest overall test performance but at a slight cost of misclassifying minority classes. We recommend training the GNN teacher with a weighted cross-entropy loss to ensure minority-class logits are not under-represented. Additionally, apply robust post-hoc calibration to further boost student performance and minority-class performance. Figure 19 shows the calibration plots before and after applying post-hoc calibration. Figure 20 shows the comparison of weighted test F1 score between

| Model | Acc_Train ± std | Acc_Val ± std | Acc_test ± std | F1_train ± std | F1_Val ± std | F1_test ± std |
|---|---|---|---|---|---|---|
| Teacher Model | 0.9982±0.0007 | 0.9707±0.0012 | 0.9685±0.0012 | 0.9982±0.0007 | 0.9707±0.0012 | 0.9685±0.0012 |
| ExtraTrees trained on hard labels | 0.8368 ± 0.0046 | 0.8337 ± 0.0024 | 0.8301 ± 0.0023 | 0.8226 ± 0.0061 | 0.8202 ± 0.0035 | 0.8126 ± 0.0028 |
| ExtraTrees trained on logits | 0.8873 ± 0.0037 | 0.8811 ± 0.0032 | 0.8817 ± 0.0063 | 0.8823 ± 0.0049 | 0.8756 ± 0.0044 | 0.8758 ± 0.0078 |
| ExtraTrees trained on Calibrated logits using IR | 0.9229 ± 0.0011 | 0.9145 ± 0.0031 | 0.9072 ± 0.0037 | 0.9210 ± 0.0012 | 0.9122 ± 0.0032 | 0.9047 ± 0.0039 |
| ExtraTrees trained on Calibrated logits using temp scaling-BS | 0.9229 ± 0.0019 | 0.9127 ± 0.0032 | 0.9102 ± 0.0029 | 0.9210 ± 0.0020 | 0.9104 ± 0.0033 | 0.9077 ± 0.0030 |
| **ExtraTrees trained on Calibrated logits using temp scaling-LL** | **0.9233 ± 0.0017** | **0.9130 ± 0.0037** | **0.9102 ± 0.0032** | **0.9215 ± 0.0018** | **0.9107 ± 0.0038** | **0.9077 ± 0.0034** |
| XGBoost trained on hard labels | 0.9194 ± 0.0000 | 0.8957 ± 0.0000 | 0.8959 ± 0.0000 | 0.9174 ± 0.0000 | 0.8928 ± 0.0000 | 0.8928 ± 0.0000 |
| XGBoost trained on logits | 0.8675 ± 0.0000 | 0.8543 ± 0.0000 | 0.8544 ± 0.0000 | 0.8580 ± 0.0000 | 0.8434 ± 0.0000 | 0.8431 ± 0.0000 |
| XGBoost trained on calibrated probs using IR | 0.9166 ± 0.0000 | 0.8935 ± 0.0000 | 0.8924 ± 0.0000 | 0.9141 ± 0.0000 | 0.8898 ± 0.0000 | 0.8885 ± 0.0000 |
| **XGBoost trained on calibrated probs using temp scaling-BS** | **0.9178 ± 0.0000** | **0.8946 ± 0.0000** | **0.8976 ± 0.0000** | **0.9154 ± 0.0000** | **0.8908 ± 0.0000** | **0.8940 ± 0.0000** |
| XGBoost trained on calibrated probs using temp scaling-LL | 0.9167 ± 0.0000 | 0.8941 ± 0.0000 | 0.8944 ± 0.0000 | 0.9142 ± 0.0000 | 0.8904 ± 0.0000 | 0.8908 ± 0.0000 |
| HistGrad trained on hard labels | 0.9549 ± 0.0002 | 0.9296 ± 0.0010 | 0.9273 ± 0.0013 | 0.9546 ± 0.0002 | 0.9287 ± 0.0011 | 0.9263 ± 0.0013 |
| HistGrad trained on logits | 0.9085 ± 0.0004 | 0.8954 ± 0.0025 | 0.8954 ± 0.0015 | 0.9053 ± 0.0004 | 0.8914 ± 0.0025 | 0.8912 ± 0.0015 |
| HistGrad trained on calibrated probs using IR | 0.9335 ± 0.0016 | 0.9164 ± 0.0042 | 0.9126 ± 0.0033 | 0.9321 ± 0.0017 | 0.9143 ± 0.0044 | 0.9103 ± 0.0036 |
| HistGrad trained on calibrated probs using temp scaling-BS | 0.9341 ± 0.0015 | 0.9171 ± 0.0017 | 0.9144 ± 0.0029 | 0.9327 ± 0.0015 | 0.9151 ± 0.0018 | 0.9122 ± 0.0032 |
| HistGrad trained on calibrated probs using temp scaling-LL | 0.9333 ± 0.0028 | 0.9162 ± 0.0045 | 0.9139 ± 0.0033 | 0.9319 ± 0.0030 | 0.9141 ± 0.0048 | 0.9117 ± 0.0036 |
| Random Forest trained on hard labels | 0.8243 ± 0.0014 | 0.8124 ± 0.0018 | 0.8133 ± 0.0006 | 0.8150 ± 0.0019 | 0.8027 ± 0.0027 | 0.8035 ± 0.0011 |
| Random Forest trained on logits | 0.8782 ± 0.0015 | 0.8668 ± 0.0032 | 0.8684 ± 0.0010 | 0.8736 ± 0.0017 | 0.8613 ± 0.0033 | 0.8631 ± 0.0011 |
| Random Forest trained on calibrated probs using IR | 0.9138 ± 0.0015 | 0.8942 ± 0.0007 | 0.8948 ± 0.0009 | 0.9117 ± 0.0017 | 0.8912 ± 0.0008 | 0.8919 ± 0.0010 |
| Random trained on calibrated probs using temp scaling-BS | 0.9133 ± 0.0019 | 0.8941 ± 0.0004 | 0.8934 ± 0.0016 | 0.9111 ± 0.0021 | 0.8910 ± 0.0006 | 0.8903 ± 0.0018 |
| **Random trained on calibrated probs using temp scaling-LL** | **0.9133 ± 0.0020** | **0.8943 ± 0.0009** | **0.8936 ± 0.0025** | **0.9112 ± 0.0022** | **0.8912 ± 0.0011** | **0.8905 ± 0.0027** |
| LightGBM trained on hard labels | 0.9537 ± 0.0000 | 0.9269 ± 0.0000 | 0.9318 ± 0.0000 | 0.9533 ± 0.0000 | 0.9258 ± 0.0000 | 0.9309 ± 0.0000 |
| LightGBM trained on logits | 0.9120 ± 0.0000 | 0.8977 ± 0.0000 | 0.8979 ± 0.0000 | 0.9089 ± 0.0000 | 0.8939 ± 0.0000 | 0.8940 ± 0.0000 |
| LightGBM trained on Calibrated probs using IR | 0.9377 ± 0.0000 | 0.9214 ± 0.0000 | 0.9176 ± 0.0000 | 0.9365 ± 0.0000 | 0.9196 ± 0.0000 | 0.9156 ± 0.0000 |
| LightGBM trained on Calibrated probs using temp scaling-BS | 0.9363 ± 0.0000 | 0.9186 ± 0.0000 | 0.9139 ± 0.0000 | 0.9350 ± 0.0000 | 0.9167 ± 0.0000 | 0.9116 ± 0.0000 |
| LightGBM trained on Calibrated probs using temp scaling-LL | 0.9373 ± 0.0000 | 0.9198 ± 0.0000 | 0.9165 ± 0.0000 | 0.9361 ± 0.0000 | 0.9179 ± 0.0000 | 0.9144 ± 0. |

**Table 23**. Model performance-CoauthorPhysics dataset. Note: The logits represent the raw outputs of the teacher model. IR denotes Isotonic Regression, BS denotes Brier score reduction, and LL denotes log loss reduction. Values in bold denote the performance of student models that learned well from the teacher model and outperformed their counterparts trained on hard labels.Std denotes the standard deviation

| Model | Number_of_Parameters | Best_Performance | DQ_Score |
|---|---|---|---|
| ExtraTrees trained on calibrated probs using LL | 952.634935 | 0.911 | 0.0310 |
| XGBoost trained on calibrated probs using BS | 1535.46 | 0.894 | 0.0403 |
| Random trained on calibrated probs using LL | 1392.373 | 0.8932 | 0.0406 |

**Table 24**. Distillation quality Scores-F1 score as the performance metric.

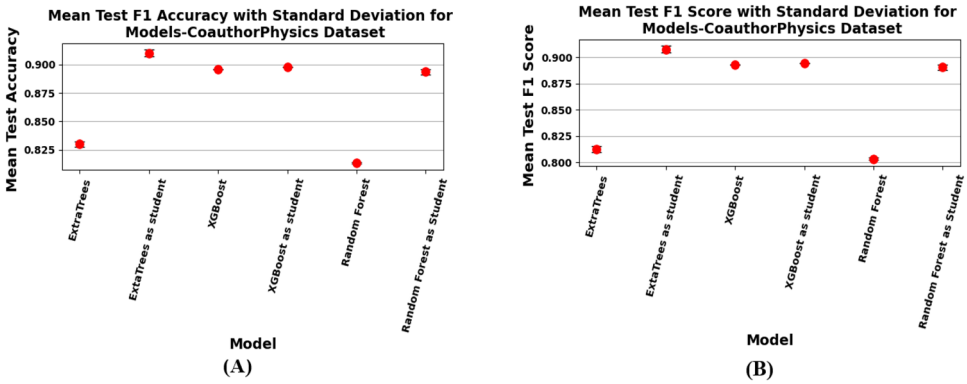


**Figure 16**. Performance of best performing student models and their counterparts on the test set-coauthorphysics. We see the student models outperforming their counterparts.

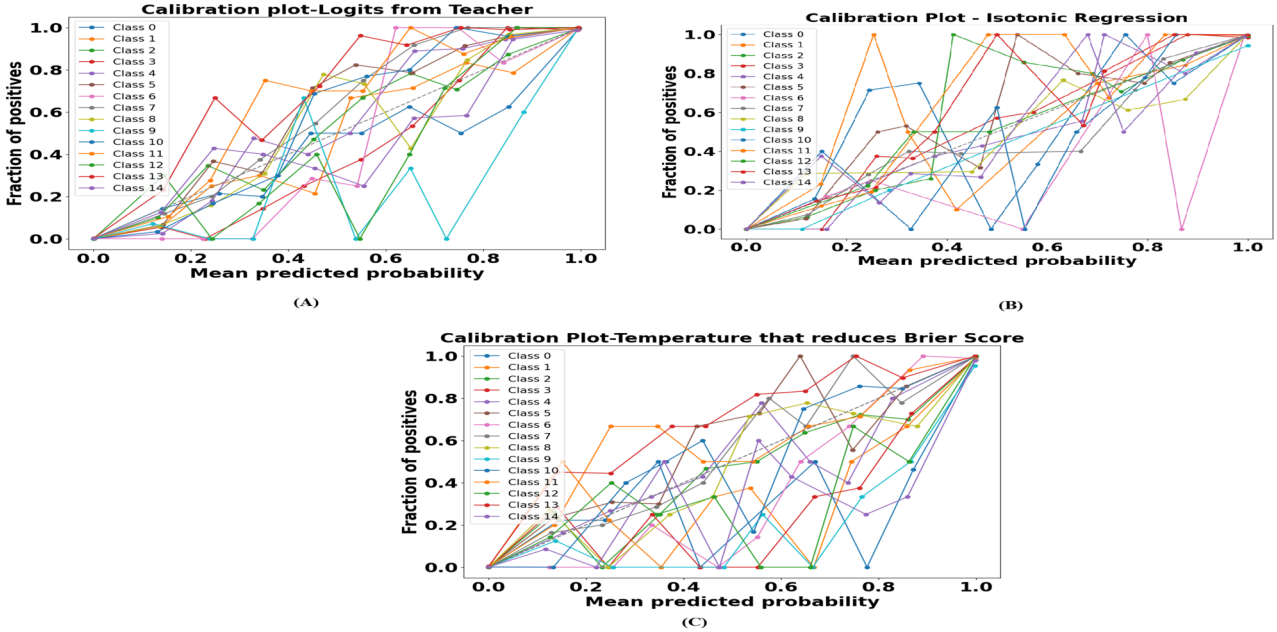


**Figure 17**. (**A**) Calibration plot: raw logits converted to probabilities. (**B**) Calibration plot after applying isotonic regression. (**C**) Calibration plot after applying temperature scaling with a temperature that reduces Stratified Brier score.

| Method/Data | Stratified Brier Score |
|---|---|
| Before Calibration | Overall: 0.0138 |
| Isotonic Regression | **Overall: 0.014** |
| Temp Scaling-Reduces Brier Score | **Overall: 0.0121** |

**Table 25**. Stratified brier scores-CoauthorCS dataset. The first overall value here is not bold. Can you please make it bold here?

| Model | Acc_Train ± std | Acc_Val ± std | Acc_test ± std | F1_train ± std | F1_Val ± std | F1_test ± std |
|---|---|---|---|---|---|---|
| Teacher Model | 0.9982±0.0005 | 0.9392±0.0006 | 0.9406±0.0039 | 0.9982±0.0005 | 0.9393±0.0007 | 0.9406±0.0040 |
| XGBoost trained on hard labels | 0.9591 ± 0.0000 | 0.8857 ± 0.0000 | 0.8664 ± 0.0000 | 0.9593 ± 0.0000 | 0.8865 ± 0.0000 | 0.8684 ± 0.0000 |
| XGBoost trained on logits | 0.8039 ± 0.0000 | 0.7657 ± 0.0000 | 0.7650 ± 0.0000 | 0.8028 ± 0.0000 | 0.7627 ± 0.0000 | 0.7633 ± 0.0000 |
| XGBoost trained on calibrated probs using IR | 0.9524 ± 0.0000 | 0.8837 ± 0.0000 | 0.8729 ± 0.0000 | 0.9521 ± 0.0000 | 0.8800 ± 0.0000 | 0.8710 ± 0.0000 |
| **XGBoost trained on calibrated probs using temp scaling-BS** | **0.9523 ± 0.0000** | **0.8843 ± 0.0000** | **0.8746 ± 0.0000** | **0.9520 ± 0.0000** | **0.8813 ± 0.0000** | **0.8728 ± 0.0000** |
| ExtraTrees trained on hard labels | 0.7617± 0.0063 | 0.7504 ±0.003 | 0.7470 ± 0.010 | 0.76490 ± 0.0064 | 0.7530 ±0.0027 | 0.7511± 0.01022 |
| ExtraTrees trained on logits | 0.8102 ± 0.0029 | 0.7918 ± 0.0054 | 0.7819 ± 0.0066 | 0.8063 ± 0.0018 | 0.7849 ± 0.0041 | 0.7811 ± 0.0058 |
| ExtraTrees trained on Calibrated probs using IR | 0.9439 ± 0.0008 | 0.8892 ± 0.0011 | 0.8805 ± 0.0033 | 0.9435 ± 0.0009 | 0.8860 ± 0.0013 | 0.8788 ± 0.0035 |
| **ExtraTrees trained on Calibrated probs using temp scaling-BS** | **0.9448 ± 0.0003** | **0.8904 ± 0.0019** | **0.8813 ± 0.0053** | **0.9444 ± 0.0003** | **0.8872 ± 0.0017** | **0.8793 ± 0.0050** |
| HistGrad trained on hard labels | 0.9758 ± 0.0005 | 0.8835 ± 0.0023 | 0.8771 ± 0.0011 | 0.9759 ± 0.0006 | 0.8844 ± 0.0026 | 0.8781 ± 0.0011 |
| HistGrad trained on logits | 0.8455 ± 0.0028 | 0.8144 ± 0.0059 | 0.7995 ± 0.0032 | 0.8448 ± 0.0027 | 0.8111 ± 0.0051 | 0.7997 ± 0.0037 |
| **HistGrad trained on calibrated probs using IR** | **0.9406 ± 0.0009** | **0.8961 ± 0.0015** | **0.8916 ± 0.0015** | **0.9399 ± 0.0010** | **0.8935 ± 0.0015** | **0.8905 ± 0.0016** |
| HistGrad trained on calibrated probs using temp scaling-BS | 0.9403 ± 0.0012 | 0.8969 ± 0.0037 | 0.8900 ± 0.0019 | 0.9396 ± 0.0012 | 0.8946 ± 0.0038 | 0.8887 ± 0.0016 |
| Random Forest trained on hard labels | 0.6541 ± 0.0049 | 0.6326 ± 0.0033 | 0.6337 ± 0.0089 | 0.6647± 0.0059 | 0.6422 ± 0.004720 | 0.64595 ± 0.0105 |
| Random Forest trained on logits | 0.7938 ± 0.0043 | 0.7724 ± 0.0027 | 0.7581 ± 0.0048 | 0.7939 ± 0.0044 | 0.7716 ± 0.0028 | 0.7607 ± 0.0000 |
| **Random Forest trained on calibrated probs using IR** | **0.9369 ± 0.0009** | **0.8759 ± 0.0020** | **0.8680 ± 0.0020** | **0.9366 ± 0.0009** | **0.8734 ± 0.0021** | **0.8659 ± 0.0020** |
| Random trained on calibrated probs using temp scaling-BS | 0.9374 ± 0.0006 | 0.8777 ± 0.0025 | 0.8675 ± 0.0008 | 0.9372 ± 0.0007 | 0.8752 ± 0.0028 | 0.8657 ± 0.0006 |
| LightGBM trained on hard labels | 0.9861 ± 0.0000 | 0.8920 ± 0.0000 | 0.8768 ± 0.0000 | 0.9861 ± 0.0000 | 0.8922 ± 0.0000 | 0.8772 ± 0.0000 |
| LightGBM trained on logits | 0.8578 ± 0.0000 | 0.8271 ± 0.0000 | 0.8086 ± 0.0000 | 0.8564 ± 0.0000 | 0.8240 ± 0.0000 | 0.8076 ± 0.0000 |
| LightGBM trained on Calibrated probs using IR | 0.9497 ± 0.0000 | 0.9007 ± 0.0000 | 0.8909 ± 0.0000 | 0.9493 ± 0.0000 | 0.8987 ± 0.0000 | 0.8896 ± 0.0000 |
| **LightGBM trained on Calibrated probs using temp scaling-BS** | **0.9473 ± 0.0000** | **0.9045 ± 0.0000** | **0.8980 ± 0.0000** | **0.9467 ± 0.0000** | **0.9027 ± 0.0000** | **0.8970 ± 0.0000** |

**Table 26**. Model performance-CoauthorCS dataset. Note: The logits represent the raw outputs of the teacher model. IR denotes Isotonic Regression, BS denotes Brier score reduction, and LL denotes log loss reduction. Values in bold denote the performance of student models that learned well from the teacher model and outperformed their counterparts trained on hard labels. Std denotes the standard deviation. Accuracy and weighted F1-scores are reported to four decimal places. Values may appear identical (especially for the teacher) due to rounding but can differ at higher precision (>6 decimal places)

| Model | Number_of_Parameters | Best_Performance | DQ_Score |
|---|---|---|---|
| ExtraTrees trained on calibrated probs using BS | 2554.529 | 0.8843 | 0.036 |
| LightGBM trained on calibrated probs using BS | 2093.5 | 0.8970 | 0.029 |
| Random trained on calibrated probs using IR | 590.5 | 0.8679 | 0.041 |
| HistGrad trained on calibrated probs using IR | 1837.45 | 0.8921 | 0.0303 |
| XGBoost trained on calibrated probs using BS | 3289.5 | 0.8728 | 0.04548 |

**Table 27**. Distillation quality scores-F1 score as the performance metric.

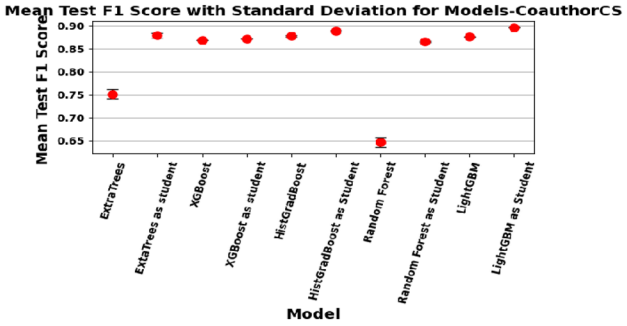


**Figure 18**. Performance of best performing student models and their counterparts on the test set-CoauthorCS. We see the student models outperforming their counterparts.

| Model | Acc_Train ± std | Acc_Val ± std | Acc_test ± std | F1_train ± std | F1_Val ± std | F1_test ± std |
|---|---|---|---|---|---|---|
| Teacher Model | 0.9971 ± 0.0015 | 0.9962 ± 0.0045 | 0.9008 ± 0.0094 | 0.9969 ± 0.0015 | 0.9959 ± 0.0045 | 0.8956 ± 0.0085 |
| ExtraTrees trained on hard labels | 0.9980 ± 0.000204 | 0.9974± 0.00031 | 0.81044 ±0.02933 | 0.99806± 0.000204 | 0.99738 ±.00031 | 0.83005 ± 0.0202 |
| **ExtraTrees trained on logits** | **0.99537 ± 0.00033** | **0.99572 ± 0.00046** | **0.89194 ± 0.01143** | **0.99520 ± 0.00033** | **0.99542 ± 0.00047** | **0.885 ± 0.01039** |
| ExtraTrees trained on Calibrated logits using IR | 0.9973 ± 0.0001 | 0.9973 ± 0.0001 | 0.7958 ± 0.0302 | 0.9971 ± 0.0001 | 0.9970 ± 0.0001 | 0.8176 ± 0.0193 |
| ExtraTrees trained on Calibrated logits using temp scaling-BS | 0.99737 ± 0.00007 | 0.99719 ± 0.00031 | 0.79556 ± 0.03426 | 0.99720 ± 0.00007 | 0.99691 ± 0.00031 | 0.81941 ± 0.02289 |
| ExtraTrees trained on Calibrated logits using temp scaling-LL | 0.9972 ± 0.0000 | 0.9970 ± 0.0001 | 0.7893 ± 0.0352 | 0.9971 ± 0.0000 | 0.9967 ± 0.0001 | 0.8128 ± 0.0221 |
| XGBoost trained on hard labels | 0.9987 ± 0.0000 | 0.9968 ± 0.0000 | 0.7491 ± 0.0000 | 0.9987 ± 0.0000 | 0.9967 ± 0.0000 | 0.7839 ± 0.0000 |
| **XGBoost trained on logits** | **0.9958 ± 0.0000** | **0.9949 ± 0.0000** | **0.8673 ± 0.0000** | **0.9956 ± 0.0000** | **0.9946 ± 0.0000** | **0.8712 ± 0.0000** |
| XGBoost trained on Calibrated logits using IR | 0.9973 ± 0.0000 | 0.9965 ± 0.0000 | 0.8306 ± 0.0000 | 0.9971 ± 0.0000 | 0.9962 ± 0.0000 | 0.8425 ± 0.0000 |
| XGBoost trained on calibrated logits using temp scaling-BS | 0.9974 ± 0.0000 | 0.9966 ± 0.0000 | 0.7498 ± 0.0000 | 0.9972 ± 0.0000 | 0.9963 ± 0.0000 | 0.7827 ± 0.0000 |
| XGBoost trained on calibrated logits using temp scaling-LL | 0.9975 ± 0.0000 | 0.9968 ± 0.0000 | 0.8031 ± 0.0000 | 0.9973 ± 0.0000 | 0.9965 ± 0.0000 | 0.8216 ± 0.0000 |
| HistGrad trained on hard labels | 0.9992 ± 0.0001 | 0.9950 ± 0.0005 | 0.7563 ± 0.0033 | 0.9990 ± 0.0001 | 0.9947 ± 0.0005 | 0.7909 ± 0.0020 |
| **HistGrad trained on logits** | **0.9951 ± 0.0004** | **0.9662 ± 0.0040** | **0.8156 ± 0.0443** | **0.9949 ± 0.0004** | **0.9676 ± 0.0036** | **0.8354 ± 0.0312** |
| HistGrad trained on calibrated probs | 0.9973 ± 0.0001 | 0.9953 ± 0.0004 | 0.8049 ± 0.0508 | 0.9971 ± 0.0001 | 0.9950 ± 0.0004 | 0.8215 ± 0.0361 |
| HistGrad trained on calibrated logits using temp scaling-BS | 0.9975 ± 0.0001 | 0.9958 ± 0.0005 | 0.7986 ± 0.0489 | 0.9974 ± 0.0001 | 0.9956 ± 0.0005 | 0.8167 ± 0.0334 |
| HistGrad trained on calibrated logits using temp scaling-LL | 0.9975 ± 0.0001 | 0.9959 ± 0.0006 | 0.8031 ± 0.0483 | 0.9973 ± 0.0001 | 0.9956 ± 0.0006 | 0.8199 ± 0.0329 |
| Random Forest trained on hard labels | 0.99934 ± 0.00013 | 0.99766±0.00031 | 0.8475 ± 0.021790 | 0.9993 ± 0.0001 | 0.99763 ± 0.00030 | 0.85659 ± 0.0166 |
| **Random Forest trained on logits** | **0.99444 ± 0.00006** | **0.99497 ± 0.00004** | **0.87928 ± 0.00717** | **0.99428 ± 0.00006** | **0.99469 ± 0.00004** | **0.87526 ± 0.00543** |
| Random Forest trained on calibrated probs using IR | 0.9978 ± 0.0000 | 0.9970 ± 0.0002 | 0.8647 ± 0.0132 | 0.9977 ± 0.0000 | 0.9967 ± 0.0002 | 0.8669 ± 0.0090 |
| Random trained on calibrated logits using temp scaling-BS | 0.9976 ± 0.0001 | 0.9969 ± 0.0001 | 0.8465 ± 0.0247 | 0.9975 ± 0.0001 | 0.9966 ± 0.0001 | 0.8538 ± 0.0172 |
| Random trained on calibrated logits using temp scaling-LL | 0.9976 ± 0.0001 | 0.9969 ± 0.0004 | 0.8346 ± 0.0225 | 0.9974 ± 0.0001 | 0.9966 ± 0.0004 | 0.8456 ± 0.0169 |
| LightGBM trained on hard labels | 0.9983 ± 0.0000 | 0.9966 ± 0.0000 | 0.8546 ± 0.0083 | 0.9983 ± 0.0000 | 0.9965 ± 0.0000 | 0.8637 ± 0.0066 |
| **LightGBM trained on logits** | **0.9951 ± 0.0000** | **0.9942 ± 0.0001** | **0.8973 ± 0.0008** | **0.9949 ± 0.0000** | **0.9939 ± 0.0001** | **0.8790 ± 0.0004** |
| LightGBM trained on Calibrated logits using IR | 0.9977 ± 0.0000 | 0.9968 ± 0.0000 | 0.8752 ± 0.0099 | 0.9975 ± 0.0000 | 0.9965 ± 0.0000 | 0.8715 ± 0.0074 |
| LightGBM trained on Calibrated logits using temp scaling-BS | 0.99762 ± 0.00000 | 0.99692 ± 0.00000 | 0.85883 ± 0.01237 | 0.99745 ± 0.00000 | 0.99663 ± 0.00000 | 0.86122 ± 0.01041 |
| LightGBM trained on Calibrated logits using temp scaling-LL | 0.9976 ± 0.0000 | 0.9968 ± 0.0000 | 0.8668 ± 0.0163 | 0.9974 ± 0.0000 | 0.9965 ± 0.0000 | 0.8657 ± 0.0111 |

**Table 28**. Model performance-synthetic dataset1. Note: The logits represent the raw outputs of the teacher model. IR denotes Isotonic Regression, BS denotes Brier score reduction, and LL denotes log loss reduction. Values in bold denote the performance of student models that learned well from the teacher model and outperformed their counterparts trained on hard labels. Std denotes the standard deviation. Weighted cross-entropy can be employed to better capture information about minority classes

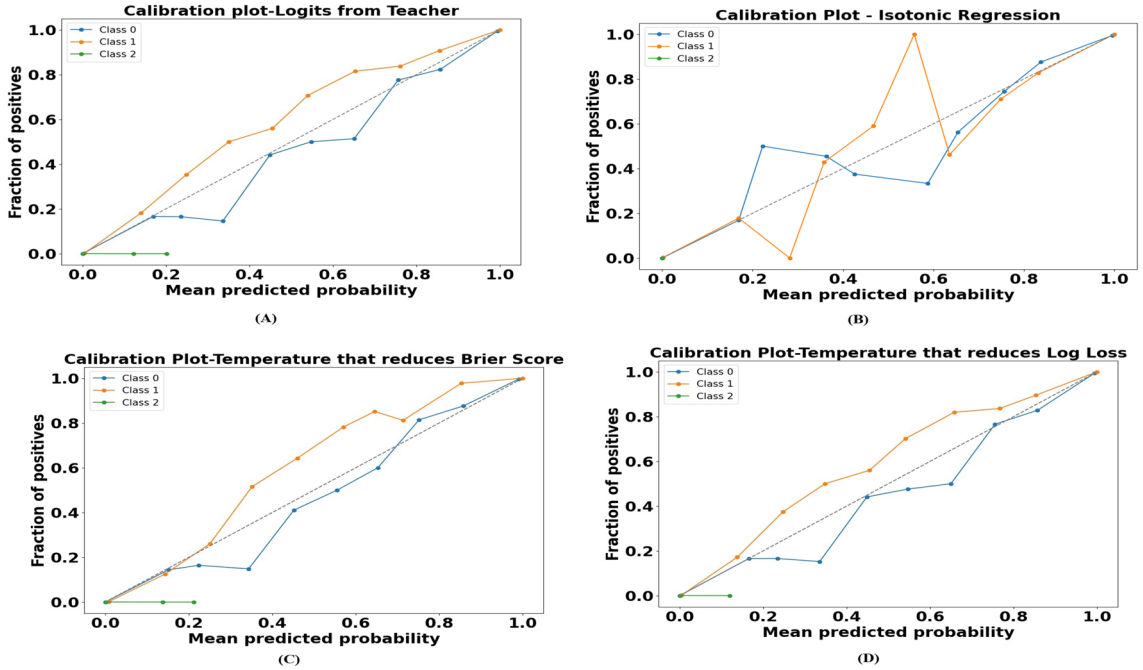


**Figure 19**. (**A**) Calibration plot: raw logits converted to probabilities. (**B**) Calibration plot after applying isotonic regression. (**C**) Calibration plot after applying temperature scaling with a temperature that reduces Brier score. (**D**) Calibration plot after applying temperature scaling with a temperature that reduces negative log-likelihood (log loss).

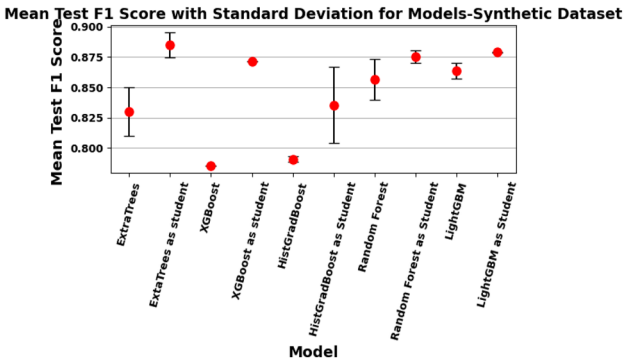


**Figure 20**. Performance of best performing student models and their counterparts on the test set-synthetic dataset. We see student models outperforming their counterparts.

models trained on hard labels and their student counterparts, which performed the best (as per the Table 28). Table 29 represents the stratified Brier scores and log loss values obtained before and after calibration. The distillation quality scores of the student models that performed the best are summarized in Table 30.

**Input:** Number of nodes $n$=60k, m=5, number of clusters $k$=3, noise factor $\delta$.
**Output:** Preprocessed feature sets $X_{train}$, $X_{val}$ and $X_{test}$, labels $y$, graph $G$.
Generate graph $G$ using the Barabási-Albert model with $n$ nodes, $m$ edges,
Compute various features such as degree, clustering coefficient, and eigenvector centrality.
Apply KMeans clustering with $k$ clusters to extracted features to generate synthetic labels $y$.
Apply the standard scaler to training and validation set. Apply Gaussian noise scaled by $\delta$ to simulate distribution shift in test set and standardize it.
**Return:** Feature subsets, $y$, $G$.

**Algorithm 2**. Feature engineering and synthetic label generation with distribution shift

| Method/data | Stratified brier score | Log loss |
|---|---|---|
| Before calibration | Class 0 :0.01275 | Class 0: 0.01597 |
| | Class 1 :0.01086 | Class 1: 0.01496 |
| | Class 2: 0.49084 | Class 2:0.00355 |
| | **Overall:0.17148** | **Overall:0.01150** |
| Isotonic regression | Class 0 : 0.0140 | Class 0: 0.0180 |
| | Class 1: 0.0136 | Class 1: 0.0170 |
| | Class 2 :0.4909 | Class 2: 0.0028 |
| | **Overall :0.1728** | **Overall:0.0126** |
| Temp scaling - reduces brier score | Class 0:0.0126 | Class 0 :0.0166 |
| | Class 1:0.0102 | Class 1 :0.0161 |
| | Class 2:0.4815 | Class 2 : 0.0040 |
| | **Overall: 0.1681** | **Overall :0.0122** |
| Temp scaling - reduces log loss | Class 0:0.01279 | Class 0:0.01595 |
| | Class 1:0.01094 | Class 1:0.01488 |
| | Class 2 :0.49160 | Class 2:0.00351 |
| | **Overall: 0.17177** | **Overall: 0.01144** |

**Table 29**. Stratified brier scores and log loss values-synthetic dataset. **Note:** The extremely low Log Loss observed for Class 2 is likely due to the small sample size for that class, which may be misleading. Isotonic Regression did not yield an improvement over the baseline calibration metrics

| Model | Number_of_Parameters | Best_Performance | DQ_Score |
|---|---|---|---|
| ExtraTrees trained on logits | 409.975 | 0.89539 | 0.0084 |
| XGBoost trained on logits | 462.97 | 0.8712 | 0.022 |
| Random trained on logits | 1331.635 | 0.8806 | 0.0248 |
| HistGrad trained on logits | 679.8 | 0.8666 | 0.02676 |
| LightGBM trained on logits | 692.15 | 0.8794 | 0.0197 |

**Table 30**. Distillation quality scores-F1 score as the performance metric.

| Dataset | Complexity of train+val graph (graph energy) | Complexity of test graph (graph energy) | Number of features | Feature type | Number of classes | Distribution shift: covariate and label shift | Did all student models benefit from KD? |
|---|---|---|---|---|---|---|---|
| Our dataset | 528099.8645 | 145243.7451519 | 32 | Spatial and Morphological | 2 | Yes (both covariate and label shift) | Yes |
| Placenta | Large for computation | Large for computation | 64 | Morphological | 9 | Yes (covariate shift) | Yes |
| BRCA-M2C | 33879.69182 | 12192.8422 | 12 | Spatial | 3 | Yes (both covariate and label shift) | Yes |
| CoauthorCS | 39464.388 | 39464.388 | 6805 | Original | 15 | No | Yes |
| CoauthorPhysics | 90782.486005 | 90782.486005 | 8415 | Original | 5 | No | No. LightGBM and HistGrad were not benefitted |
| Synthetic dataset | Large for computation | Large for computation | 7 | Topological | 3 | Yes (Covariate Shift) | Yes |

**Table 31**. Factors considered to evaluate the efficacy of our proposed method.

## Factors influencing the efficacy of our approach across datasets

We considered various factors impacting our approach and tabulated them in Table 31. However, the graph complexity (which, in our case is equivalent to graph energy) could not be computed for the placenta dataset as it contained millions of nodes within a single cell graph. Similarly, the synthetic dataset also had a large number of nodes within a single graph, making complexity computation infeasible. Approximating graph complexity for very large graphs is an avenue for future work.

## Effectiveness of our approach: successes, limitations, and when it might not be too useful

Our approach proved particularly beneficial in complex scenarios involving data distribution shifts. In such cases, the logits from the teacher GNN provided richer insights than the hard labels. In addition to the knowledge transferred from the teacher to the student, it also helped curb overfitting to the training data, preventing student models from becoming overly specialized on the training set. In our experiments, student models, like Random Forest or ExtraTrees, benefited constantly from the GNN's logits. These models were able to leverage the rich

information encoded in the logits to make more accurate predictions. The performance of boosting models varied across different scenarios. Specifically, when the test distribution closely mirrored the training distribution, slight overfitting of these models to the training data proved beneficial. Bagging models demonstrated more consistent improvements with knowledge distillation across different datasets and complexities, making them a more favorable choice for distillation. In simpler cases, where data relationships are primarily linear or the graph has very low complexity, knowledge distillation from GNN becomes less impactful. In such scenarios, simpler models can directly utilize the cell graph features for classification, achieving effective results without requiring a teacher GNN. Introducing a GNN in these cases adds unnecessary complexity. This aligns with the findings of the authors in[34], who demonstrated that a simple classifier using the 15 most predictive feature-driven local cell graph features identified via the Wilcoxon Rank Sum Test (WRST) achieved an average AUC of 0.68, thereby outperforming a deep learning model.

## Discussion and major takeaways

To address the first question regarding the benefits of knowledge distillation from the teacher GNN, we analyzed the performance of the teacher and student models under varying dataset complexities. The proposed approach was instrumental in scenarios where a distribution shift existed in the data. In such cases, student models trained on logits consistently outperformed their counterparts trained on hard labels. However, the results were mixed for non-cell graph-based datasets without distribution shifts. Bagging models improved when trained on calibrated probs, while some boosting models performed better with hard labels. We hypothesize that logits offer more effective guidance during distribution shifts than hard labels. Furthermore, logits acted as a form of regularization, helping to prevent models from overfitting to the training data, as evidenced by the results on cell graph-based datasets. Achieving high-quality logits required a sufficiently deep teacher model.

To address the second question, we observed notable differences in the feature importance assigned by models trained on hard labels versus those guided by teacher logits. For the TB dataset, the teacher-guided student model emphasized morphological features, such as contrast and circularity. Pathologists commonly use these to differentiate between AFB and the nucleus of activated macrophages. Similarly, for the BRAC_M2C dataset, the teacher-guided student model prioritized features like node clustering, reflecting the biological behavior of breast cancer cells, which tend to form tight clusters and adhere via adhesion molecules.

To answer the third question, we observed performance improvements in student models when they were trained using the ensembled outputs of the teacher model and the best-performing student model, particularly in specific datasets. These improvements were more pronounced when the best-performing student shared some architectural similarities with other student models. For instance, the performance of HistGradientBooster improved when guided by the ensembled logits of LightGBM and CG-JKNN, compared to its performance when trained solely by CG-JKNN. However, it is essential to note that the ensembled outputs were not universally beneficial. Some student models experienced a drop in performance. Also, some models did not prefer learning from the best-performing student, as they assigned a zero weight to its output.

Regarding the fourth question, teacher logit calibration provided better guidance to student models than using hard labels in most cases. In datasets like the placenta dataset, where the number of samples per class was small, isotonic regression led to lower performance and, in some instances, performed worse than using hard labels.

We hypothesize that the success of our approach stems from the student's inherent inductive bias, which functions as a powerful regularizing filter. Unlike a flexible NN student, which can overfit to the teacher's entire output function, including its flaws, a tree-based model's structural rigidity prevents it from replicating these complex, spurious correlations. This inherent limitation forces the student to approximate the teacher's knowledge using simpler, rule-based tools, thereby capturing the dominant, generalizable signals while ignoring high-frequency noise. Further experiments are needed across varied datasets and model architectures to validate the robustness and scope of this hypothesis fully.

The major takeaways are as follows:

- Our goal was not to benchmark the teacher model against the baseline performances reported for each dataset. Instead, our primary focus was demonstrating the efficacy of using the teacher's logits as a supervisory signal for training student models.
- Our approach using teacher GNN logits improved student model performance under distribution shifts by capturing model uncertainty and relative class similarities, which in turn revealed subtle transitional states in cellular morphology that hard labels may obscure. For example, in the placenta dataset, the student model produces very similar confidence scores for class 1 and class 2, suggesting that these classes may share morphological features during transformation[129,130]. Additional details can be found in our supplementary files. However, further expert evaluation is necessary to determine whether these outputs genuinely represent biological transitions or if they instead reflect limitations in feature extraction. Moreover, the teacher may produce noisy logits for classes with few samples due to insufficient representation learning.
- Bagging models consistently benefited from using logits compared to hard labels. In contrast, boosting models showed mixed results, with some cases favoring hard labels over logits, especially in datasets with no distribution shift. During our experiments on the CoauthorCS network, we found that bagging models, such as Random Forest, performed well with calibrated probabilities obtained through isotonic regression. These probabilities focused on improving reliability, even though this came at the cost of resolution. On the other hand, booster models, such as XGBoost, performed better with calibrated probabilities obtained through temperature scaling, which provided higher resolution but slightly less reliability compared to isotonic regression. We believe this difference is related to the way these algorithms function.

- We observed that the teacher-guided student model placed greater emphasis on morphological features for the TB dataset than its counterpart trained on hard labels. This suggests that combining local cell graph features and morphological features provides better guidance and performance than using either morphological or local cell graph features alone. We believe that the student model can serve as a partial proxy for understanding which features the teacher considers important.
- While using weighted cross-entropy loss helps address the class imbalance, it does not tackle calibration issues. A more advanced loss function, such as the one proposed in[121], could be employed to handle both class imbalance and calibration simultaneously.

The focus of this work was to demonstrate that it is possible to distill knowledge from neural to non-neural network models as students and that these simpler models can also learn effectively from the logits of a teacher GNN. Even when we observe a performance gap between the teacher and student models (in the case of the TB dataset), often due to the teacher's use of graph structure, the results indicate that the distilled logits provide better guidance than hard labels. This opens up opportunities for further improvements, such as incorporating intermediate teacher embeddings with node features, to help the students better approximate the teacher's full capabilities. In our study, we deliberately chose non-neural student models for several reasons. Their enhanced interpretability is a primary advantage. Tree-based ensembles enable the straightforward extraction of decision rules (compared to GNNs), making model decisions easily understandable, an essential aspect in applications involving TB, Placenta, and Breast Cancer datasets. Moreover, prior work, such as that by Frosst and Hinton[131], has shown that distilling a deep neural network into a simpler decision tree can improve the tree's performance compared to training on hard labels alone. By transferring the teacher's rich, implicit relational knowledge to these students via its logits, we allow them to effectively operate using only cell-level feature vectors (which include morphological and local cell graph features), thereby broadening applicability to scenarios where full-cell graphs are unavailable. Their simplicity also offers multiple practical advantages: they require significantly less hyperparameter tuning[132], are easier to implement, and their decision boundaries are more readily visualized compared to more complex methods, such as those employed in approaches like GNNBoundary[133]. Additionally, the differences in feature importance between the student models and counterparts trained on hard labels provide valuable insights into the teacher's decision-making process. Additionally, successfully transferring the teacher's knowledge to a non-neural model demonstrates that these valuable insights are not exclusive to neural architectures but can be effectively captured by different function approximations, underscoring its generality.

## Limitations of our work

Our work primarily focused on node-level classification, which limits its applicability to graph-level classification tasks. In this study, all interactions between cells were assigned equal weight (weight=1). However, specific interactions may be more biologically significant than others. For example, interactions between lymphocytes and cancer cells could have a stronger impact on disease progression than other cell-cell interactions. While we employed calibration methods such as isotonic regression and temperature scaling, we did not explore other popular techniques. Specifically, for multiclass calibration, we adapted isotonic regression using a one-vs-all approach, which may not fully capture the subtleties of multiclass classification compared to methods specifically designed for this purpose, such as Matrix Scaling, Vector Scaling[111], or Dirichlet Calibration[115]. The calibration performance was assessed using the Stratified Brier Score and Log Loss. However, our analysis may lack comprehensiveness as metrics like Expected Calibration Error (ECE)[134] were not considered. Moreover, we did not explore pre-calibration techniques that integrate temperature learning during GNN training to generate pre-calibrated probabilities, leaving the effectiveness of the pre-calibrated softmax probabilities unexamined. We also did not assess whether the student models' predictions were calibrated. While we used weighted F1 score and accuracy as our primary performance metrics, our analysis could be enhanced by incorporating other performance measures that provide broader insights. While weighted F1 shows overall gains from distillation, the teacher's logits remain noisy for the very rare classes, so improvements are uneven and some classes might not see a clear benefit. Furthermore, while we focused on evaluating the generalization capabilities of student models trained with teacher logits, we did not analyze fidelity[17,135], which is a measure of how closely the student models replicate the teacher's outputs. Incorporating fidelity in future evaluations could provide a more comprehensive understanding of the trade-offs between generalization and fidelity.Additionally, to fully validate our findings, the framework should be evaluated under more rigorous shift conditions.

## Conclusion and future work

We explored logit-based knowledge distillation from GNNs trained on cell graphs to non-neural student models. The study assessed the efficacy of this approach under different dataset complexities, including factors such as varying graph complexity and the presence or absence of distribution shifts. Our approach proved particularly beneficial when the test distribution differed from the training distribution. The rich information embedded in the logits and their regularization effect benefited the student models. Additionally, we investigated the scenarios where the calibration of logits could enhance student performance. Post-hoc calibration demonstrated its utility when ample samples were available in each class and when there was no distribution shift. Bagging models consistently benefited from logits, whereas booster models exhibited variable performance based on the presence or absence of shift.

We plan to experiment with other teacher models in future work, such as the Simple Graph Convolution (SGC) proposed in[136]. This model aims to reduce the excess complexities typically associated with GCNs by removing intermediate non-linear transformations while still leveraging the graph structure for learning. This would also open new avenues for utilizing simpler linear models as student models, potentially reducing the overall model complexity while maintaining or improving performance. We also plan to focus on measuring

the fidelity of the student models[17]. Fidelity, in this context, refers to the ability of the student models to match the teacher's predictions across various graph datasets. We do not explicitly evaluate whether our student models are themselves well-calibrated. The calibration of these models remains an important direction for future investigation. We aim to incorporate the loss function proposed in[121], which uses a dynamic weighting factor that adjusts during the training process of our teacher GNN. This approach addresses the training bias in imbalanced datasets while improving confidence calibration. Future direction could also explore methods for training the teacher GNN to yield logits that provide a more uniformly beneficial and balanced learning signal for all classes, especially under extreme imbalance and potential distribution shifts. We also propose experimenting with synthetic datasets that do not exhibit distribution shifts but are designed to emulate the distributions of real-world networks. By extracting local graph features from these datasets, it would be intriguing to investigate whether logits offer greater utility in guiding student models. Future research could focus on designing teacher models such as $H_2$GCN[120], capable of effectively learning in both homophilic and heterophilic contexts. Advanced approaches to measure the degree of non-linearity in datasets can be employed. For example, the method described in[137] quantifies the degree of non-linearity between variables by defining the exposure of one variable to another. Synthetic dataset generators, such as ShapeGGen[128], can be employed to automatically create a variety of benchmark datasets with varying properties to evaluate the efficacy of knowledge distillation. Another interesting future direction is exploring causal knowledge distillation, where we generate causal graphs of cell graph features to guide the distillation process. Future work could explore using teacher logits as "pseudo-labels" in semi-supervised learning to provide soft targets for student models when labeled data is scarce. Visualizing the decision boundaries of student models in knowledge distillation scenarios could offer valuable insights into how these models approximate the behavior of teacher models.

## Data availability

The datasets analyzed during the current study are available in the Placenta repository (GitHub link: https://github.com/nellaker-group/placenta) and the Dataset-BRCA-M2C repository (GitHub link: https://github.com/TopoXLab/Dataset-BRCA-M2C). Similarly, the CoAuthorship networks utilized in this study are publicly available in (https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.datasets.Coauthor.html). The whole-slide images (WSI) used in the TB dataset will be made available upon request to the corresponding author. The codes used to perform the experiments and generate the results in this study is publicly available in a repository with the link (Link: https://github.com/VasundharaAcharya/Code_Knowledge_Distillation.git)

Received: 6 February 2025; Accepted: 25 July 2025
Published online: 10 August 2025

## References

1. Yener, B. Cell-graphs: Image-driven modeling of structure-function relationship. *Commun. ACM* **60**, 74–84 (2016).
2. Hinck, L. & Näthke, I. Changes in cell and tissue organization in cancer of the breast and colon. *Curr. Opin. Cell Biol.* **26**, 87–95 (2014).
3. World Health Organization, *Global Tuberculosis Report 2024* (World Health Organization, 2024).
4. Turner, R. D. et al. Tuberculosis infectiousness and host susceptibility. *J. Infect. Dis.* **216**, S636–S643 (2017).
5. Qiu, X. et al. Spatial transcriptomic sequencing reveals immune microenvironment features of *Mycobacterium tuberculosis* granulomas in lung and omentum. *Theranostics* **14**, 6185 (2024).
6. Ndlovu, H. & Marakalala, M. J. Granulomas and inflammation: Host-directed therapies for tuberculosis. *Front. Immunol.* **7**, 434 (2016).
7. Schluger, N. W. The acid-fast bacilli smear: Hail and farewell. *Am. J. Respir. Crit. Care Med.* **199**(6), 691–692 (2019).
8. Weers, A. et al. From pixels to histopathology: A graph-based framework for interpretable whole slide image analysis. arXiv preprint arXiv:2503.11846 (2025).
9. Lerner, T. R. et al. *Mycobacterium tuberculosis* cords within lymphatic endothelial cells to evade host immunity. *JCI Insight* **5**, e136937 (2020).
10. Warrender, C., Forrest, S. & Koster, F. Modeling intercellular interactions in early mycobacterium infection. *Bull. Math. Biol.* **68**, 2233–2261 (2006).
11. Janiszewska, M., Primi, M. C. & Izard, T. Cell adhesion in cancer: Beyond the migration of single cells. *J. Biol. Chem.* **295**, 2495–2505 (2020).
12. Xu, K. et al. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, 5453–5462 (PMLR, 2018).
13. Ojha, U., Li, Y., Sundara Rajan, A., Liang, Y. & Lee, Y. J. What knowledge gets distilled in knowledge distillation?. *Adv. Neural. Inf. Process. Syst.* **36**, 11037–11048 (2023).
14. Hu, C. et al. Teacher-student architecture for knowledge distillation: A survey. arXiv preprint arXiv:2308.04268 (2023).
15. Hinton, G., Vinyals, O. & Dean, J. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015).
16. Xie, Q., Luong, M.-T., Hovy, E. & Le, Q. V. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10687–10698 (2020).
17. Stanton, S., Izmailov, P., Kirichenko, P., Alemi, A. A. & Wilson, A. G. Does knowledge distillation really work?. *Adv. Neural. Inf. Process. Syst.* **34**, 6906–6919 (2021).
18. Ba, J. & Caruana, R. Do deep nets really need to be deep? *Adv. Neural Inf. Process. Syst.***27** (2014).
19. Fukui, S., Yu, J. & Hashimoto, M. Distilling knowledge for non-neural networks. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1411–1416 (IEEE, 2019).
20. Nair, A. et al. A graph neural network framework for mapping histological topology in oral mucosal tissue. *BMC Bioinform.* **23**, 506 (2022).
21. Paul, S., Yener, B. & Lund, A. W. C2P-GCN: Cell-to-patch graph convolutional network for colorectal cancer grading. In *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 1–4 (IEEE, 2024).
22. Baranwal, M., Krishnan, S., Oneka, M., Frankel, T. & Rao, A. CGAT: Cell graph attention network for grading of pancreatic disease histology images. *Front. Immunol.* **12**, 727610 (2021).
23. Zhou, Y. et al. CGC-Net: Cell graph convolutional network for grading of colorectal cancer histology images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019).

24. Su, Y., Bai, Y., Zhang, B., Zhang, Z. & Wang, W. HAT-Net: A hierarchical transformer graph neural network for grading of colorectal cancer histology images. In *Proc. Brit. Mach. Vis. Conf.* (2021).
25. Bilgin, C., Demir, C., Nagi, C. & Yener, B. Cell-graph mining for breast tissue modeling and classification. In *2007 29th Annual international conference of the IEEE Engineering in Medicine and Biology Society*, 5311–5314 (IEEE, 2007).
26. Bhattacharyya, D., Pal, A. J. & Kim, T.-H. Cell-graph coloring for cancerous tissue modelling and classification. *Multimed. Tools Appl.* **66**, 229–245 (2013).
27. Gunduz-Demir, C. Mathematical modeling of the malignancy of cancer using graph evolution. *Math. Biosci.* **209**, 514–527 (2007).
28. Gunduz, C., Yener, B. & Gultekin, S. H. The cell graphs of cancer. *Bioinformatics* **20**, i145–i151 (2004).
29. Demir, C., Gultekin, S. H. & Yener, B. Augmented cell-graphs for automated cancer diagnosis. *Bioinformatics* **21**, ii7–ii12 (2005).
30. Gunduz-Demir, C., Kandemir, M., Tosun, A. B. & Sokmensuer, C. Automatic segmentation of colon glands using object-graphs. *Med. Image Anal.* **14**, 1–12 (2010).
31. Lou, W., Li, G., Wan, X. & Li, H. Cell graph transformer for nuclei classification. *In Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 3873–3881 (2024).
32. Wang, T., Bai, J. & Nabavi, S. Single-cell classification using graph convolutional networks. *BMC Bioinform.* **22**, 1–23 (2021).
33. Wang, Y. et al. Cell graph neural networks enable the precise prediction of patient survival in gastric cancer. *NPJ Precis. Oncol.* **6**, 1–12 (2022).
34. Lu, C. et al. Feature driven local cell graph (FEDEG): predicting overall survival in early stage lung cancer. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part II 11*, 407–416 (Springer, 2018).
35. Pati, P. et al. Hierarchical graph representations in digital pathology. *Med. Image Anal.* **75**, 102264 (2022).
36. Hsu, Y.-C., Smith, J., Shen, Y., Kira, Z. & Jin, H. A closer look at knowledge distillation with features, logits, and gradients. arXiv preprint arXiv:2203.10163 (2022).
37. Sun, W. et al. Knowledge distillation with refined logits. arXiv preprint arXiv:2408.07703 (2024).
38. Yang, Y., Qiu, J., Song, M., Tao, D. & Wang, X. Distilling knowledge from graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7074–7083 (2020).
39. Kim, J., Jung, J. & Kang, U. Compressing deep graph convolution network with multi-staged knowledge distillation. *PLoS ONE* **16**, e0256187 (2021).
40. Jing, Y., Yang, Y., Wang, X., Song, M. & Tao, D. Amalgamating knowledge from heterogeneous graph neural networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15704–15713. https://doi.org/10.1109/CVPR46437.2021.01545 (2021).
41. Antaris, S. & Rafailidis, D. Distill2Vec: Dynamic graph representation learning with knowledge distillation. *CoRR*. arXiv:2011.05664 (2020).
42. Yan, B., Wang, C., Guo, G. & Lou, Y. TinyGNN: Learning efficient graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20, 1848–1856. https://doi.org/10.1145/3394486.3403236 (Association for Computing Machinery, 2020).
43. Joshi, C. K., Liu, F., Xun, X., Lin, J. & Foo, C. On representation knowledge distillation for graph neural networks. *CoRR*. arXiv:2111.04964 (2021).
44. Yang, Y., Qiu, J., Song, M., Tao, D. & Wang, X. Distilling knowledge from graph convolutional networks. *CoRR*. arXiv:2003.10477 (2020).
45. He, H., Wang, J., Zhang, Z. & Wu, F. Compressing deep graph neural networks via adversarial knowledge distillation. arXiv:2205.11678 (2022).
46. Dong, Y. et al. Reliant: Fair knowledge distillation for graph neural networks. arXiv:2301.01150 (2023).
47. Tian, Y., Xu, S. & Li, M. Decoupled graph knowledge distillation: A general logits-based method for learning MLPs on graphs. *Neural Netw.* **179**, 106567 (2024).
48. Wu, L., Lin, H., Huang, Y. & Li, S. Z. Knowledge distillation improves graph structure augmentation for graph neural networks. *Adv. Neural. Inf. Process. Syst.* **35**, 11815–11827 (2022).
49. Huo, C. et al. T2-GNN: Graph neural networks for graphs with incomplete features and structure via teacher-student distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, 4339–4346 (2023).
50. Acharya, V., Choi, D., Yener, B. & Beamer, G. Prediction of tuberculosis from lung tissue images of diversity outbred mice using jump knowledge based cell graph neural network. *IEEE Access* **12**, 17164–17194 (2024).
51. Vanea, C. et al. A new graph node classification benchmark: Learning structure from histology cell graphs. arXiv preprint arXiv:2211.06292 (2022).
52. Abousamra, S. et al. Multi-class cell detection using spatial context representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4005–4014 (2021).
53. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
54. Miao, J. & Zhu, W. Precision-recall curve (PRC) classification trees. *Evol. Intell.* **15**, 1545–1569 (2022).
55. Studer, L., Wallau, J., Dawson, H., Zlobec, I. & Fischer, A. Classification of intestinal gland cell-graphs using graph neural networks. In *2020 25th International conference on pattern recognition (ICPR)*, 3636–3643 (IEEE, 2021).
56. McKeen-Polizzotti, L. et al. Quantitative metric profiles capture three-dimensional temporospatial architecture to discriminate cellular functional states. *BMC Med. Imaging* **11**, 1–14 (2011).
57. Lerner, T. R. et al. *Mycobacterium tuberculosis* cording in the cytosol of live lymphatic endothelial cells. *bioRxiv* 595173 (2019).
58. Eppstein, D., Paterson, M. S. & Yao, F. F. On nearest-neighbor graphs. *Discrete Comput. Geom.* **17**, 263–282 (1997).
59. Guibas, L. J., Knuth, D. E. & Sharir, M. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica* **7**, 381–413 (1992).
60. Hagberg, A., Swart, P. J. & Schult, D. A. Exploring network structure, dynamics, and function using networkX. Tech. Rep. (Los Alamos National Laboratory (LANL), 2008).
61. Sarıgün, A. & Rifaioglu, A. S. Multi-mask aggregators for graph neural networks. In *The First Learning on Graphs Conference* (2022).
62. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018).
63. Brody, S., Alon, U. & Yahav, E. How attentive are graph attention networks? arXiv preprint arXiv:2105.14491 (2021).
64. Zhou, K. et al. Dirichlet energy constrained learning for deep graph neural networks. *Adv. Neural. Inf. Process. Syst.* **34**, 21834–21846 (2021).
65. Hasanzadeh, A. et al. Bayesian graph neural networks with adaptive connection sampling. In *International Conference on Machine Learning*, 4094–4104 (PMLR, 2020).
66. Rusch, T. K., Chamberlain, B., Rowbottom, J., Mishra, S. & Bronstein, M. Graph-coupled oscillator networks. In *International Conference on Machine Learning*, 18888–18909 (PMLR, 2022).
67. Liu, M., Gao, H. & Ji, S. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 338–348 (2020).
68. Rong, Y., Huang, W., Xu, T. & Huang, J. DropEdge: Towards deep graph convolutional networks on node classification. arXiv preprint arXiv:1907.10903 (2019).

69. Kim, T., Oh, J., Kim, N., Cho, S. & Yun, S.-Y. Comparing Kullback–Leibler divergence and mean squared error loss in knowledge distillation. arXiv preprint arXiv:2105.08919 (2021).
70. Ye, J. On measuring and correcting the effects of data mining and model selection. *J. Am. Stat. Assoc.* **93**, 120–131 (1998).
71. Elder, J. F. IV. The generalization paradox of ensembles. *J. Comput. Graph. Stat.* **12**, 853–864 (2003).
72. Wang, H., Huang, B. & Wang, J. Predict long-range enhancer regulation based on protein–protein interactions between transcription factors. *Nucleic Acids Res.* **49**, 10347–10368 (2021).
73. Hauenstein, S., Wood, S. N. & Dormann, C. F. Computing AIC for black-box models using generalized degrees of freedom: A comparison with cross-validation. *Commun. Stat.-Simul. Comput.* **47**, 1382–1396 (2018).
74. Rao, A. S. S. & Rao, C. R. *Principles and Methods for Data Science* (Elsevier, 2020).
75. Chakrabarti, A. & Ghosh, J. K. AIC, BIC and recent advances in model selection. In *Philosophy of Statistics* 583–605 (2011).
76. Matsuki, K., Kuperman, V. & Van Dyke, J. A. The random forests statistical technique: An examination of its value for the study of reading. *Sci. Stud. Read.* **20**, 20–33 (2016).
77. Wen, P.-J. & Huang, C. Machine learning and prediction of masked motors with different materials based on noise analysis. *IEEE Access* **10**, 75708–75719 (2022).
78. Stone, M. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J. R. Stat. Soc. Ser. B (Methodol.)* **39**, 44–47 (1977).
79. Alkhulaifi, A., Alsahli, F. & Ahmad, I. Knowledge distillation in deep learning and its applications. *PeerJ Comput. Sci.* **7**, e474 (2021).
80. Chebotar, Y. & Waters, A. Distilling knowledge from ensembles of neural networks for speech recognition. In *Interspeech*, 3439–3443 (2016).
81. Wu, C., Wu, F., Qi, T. & Huang, Y. Unified and effective ensemble knowledge distillation. arXiv preprint arXiv:2204.00548 (2022).
82. Radosavovic, I., Dollár, P., Girshick, R., Gkioxari, G. & He, K. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4119–4128 (2018).
83. Lin, T., Kong, L., Stich, S. U. & Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Adv. Neural. Inf. Process. Syst.* **33**, 2351–2363 (2020).
84. Kang, J. & Gwak, J. Ensemble learning of lightweight deep learning models using knowledge distillation for image classification. *Mathematics* **8**, 1652 (2020).
85. Liu, X., He, P., Chen, W. & Gao, J. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. arXiv preprint arXiv:1904.09482 (2019).
86. Wang, Z., Li, B., Liu, N., Wu, B. & Zhu, X. Distilling knowledge from an ensemble of convolutional neural networks for seismic fault detection. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2020).
87. Wu, C., Wu, F. & Huang, Y. One teacher is enough? Pre-trained language model distillation from multiple teachers. arXiv preprint arXiv:2106.01023 (2021).
88. Chebotar, Y. & Waters, A. Distilling knowledge from ensembles of neural networks for speech recognition. In *Interspeech*, 3439–3443 (2016).
89. Du, S. et al. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *Adv. Neural. Inf. Process. Syst.* **33**, 12345–12355 (2020).
90. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.***30** (2017).
91. Kursa, M. B. & Rudnicki, W. R. Feature selection with the Boruta package. *J. Stat. Softw.* **36**, 1–13 (2010).
92. Candes, E., Fan, Y., Janson, L. & Lv, J. Panning for gold: 'model-x' knockoffs for high dimensional controlled variable selection. *J. R. Stat. Soc. Ser. B Stat Methodol.* **80**, 551–577 (2018).
93. Wallace, M. L. et al. Use and misuse of random forest variable importance metrics in medicine: Demonstrations through incident stroke prediction. *BMC Med. Res. Methodol.* **23**, 144 (2023).
94. Huang, T., You, S., Wang, F., Qian, C. & Xu, C. Knowledge distillation from a stronger teacher. *Adv. Neural. Inf. Process. Syst.* **35**, 33716–33727 (2022).
95. Romero, A. et al. FitNets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014).
96. Mirzadeh, S. I. et al. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 5191–5198 (2020).
97. Mowshowitz, A. & Dehmer, M. Entropy and the complexity of graphs revisited. *Entropy* **14**, 559–570 (2012).
98. Mezić, I., Fonoberov, V. A., Fonoberova, M. & Sahai, T. Spectral complexity of directed graphs and application to structural decomposition. *Complexity* **2019**, 9610826 (2019).
99. Pugliese, A. & Nilchiani, R. Developing spectral structural complexity metrics. *IEEE Syst. J.* **13**, 3619–3626 (2019).
100. Yao, H. et al. Wild-time: A benchmark of in-the-wild distribution shift over time. *Adv. Neural. Inf. Process. Syst.* **35**, 10309–10324 (2022).
101. Zhang, H., Singh, H., Ghassemi, M. & Joshi, S. Why did the model fail? Attributing model performance changes to distribution shifts. arXiv preprint arXiv:2210.10769 (2022).
102. Roland, T. et al. Domain shifts in machine learning based COVID-19 diagnosis from blood tests. *J. Med. Syst.* **46**, 23 (2022).
103. Nair, N. G., Satpathy, P., Christopher, J. et al. Covariate shift: A review and analysis on classifiers. In *2019 Global Conference for Advancement in Technology (GCAT)*, 1–6 (IEEE, 2019).
104. Armstrong, R. A. When to use the Bonferroni correction. *Ophthalmic Physiol. Opt.* **34**, 502–508 (2014).
105. McHugh, M. L. The Chi-square test of independence. *Biochem. Med.* **23**, 143–149 (2013).
106. Teixeira, L., Jalaian, B. & Ribeiro, B. Are graph neural networks miscalibrated? arXiv preprint arXiv:1905.02296 (2019).
107. Aggarwal, U., Popescu, A., Belouadah, E. & Hudelot, C. A comparative study of calibration methods for imbalanced class incremental learning. *Multimed. Tools Appl.* **81**, 19237–19256 (2022).
108. Rajaraman, S., Ganesan, P. & Antani, S. Deep learning model calibration for improving performance in class-imbalanced medical image classification tasks. *PLoS ONE* **17**, e0262838 (2022).
109. Yang, R., Wu, T. & Yang, Y. Loca: Logit calibration for knowledge distillation. arXiv preprint arXiv:2409.04778 (2024).
110. Wang, X., Liu, H., Shi, C. & Yang, C. Be confident! towards trustworthy graph neural networks via confidence calibration. *Adv. Neural. Inf. Process. Syst.* **34**, 23768–23779 (2021).
111. Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330 (PMLR, 2017).
112. Kuleshov, V., Fenner, N. & Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2796–2804 (PMLR, 2018).
113. Zhang, J., Kailkhura, B. & Han, T. Y.-J. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning*, 11117–11128 (PMLR, 2020).
114. Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330 (PMLR, 2017).
115. Kull, M. *et al.* Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. *Adv. Neural Inf. Process. Syst.***32** (2019).
116. Zadrozny, B. & Elkan, C. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 694–699 (2002).

117. Ovadia, Y. et al. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Adv. Neural Inf. Process. Syst.***32** (2019).
118. Paszke, A. et al. Automatic differentiation in PyTorch (2017).
119. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2623–2631 (2019).
120. Zhu, J. et al. Beyond homophily in graph neural networks: Current limitations and effective designs. *Adv. Neural. Inf. Process. Syst.* **33**, 7793–7804 (2020).
121. Fernando, K. R. M. & Tsokos, C. P. Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **33**, 2940–2951 (2021).
122. Huang, L., Zhao, J., Zhu, B., Chen, H. & Broucke, S. V. An experimental investigation of calibration techniques for imbalanced data. *IEEE Access* **8**, 127343–127352 (2020).
123. Obadinma, S., Guo, H. & Zhu, X. Class-wise calibration: A case study on COVID-19 hate speech. In *Canadian AI* (2021).
124. Zhang, H. et al. Spatial positioning of immune hotspots reflects the interplay between B and T cells in lung squamous cell carcinoma. *Cancer Res.* **83**, 1410–1425 (2023).
125. Clark, A. G. & Vignjevic, D. M. Modes of cancer cell invasion and the role of the microenvironment. *Curr. Opin. Cell Biol.* **36**, 13–22 (2015).
126. Sun, X.-Y. et al. Prognostic value and distribution pattern of tumor infiltrating lymphocytes and their subsets in distant metastases of advanced breast cancer. *Clin. Breast Cancer* **24**, e167–e176 (2024).
127. Bates, J. P., Derakhshandeh, R., Jones, L. & Webb, T. J. Mechanisms of immune evasion in breast cancer. *BMC Cancer* **18**, 1–14 (2018).
128. Agarwal, C., Queen, O., Lakkaraju, H. & Zitnik, M. Evaluating explainability for graph neural networks. *Sci. Data* **10**, 144 (2023).
129. Boss, A. L., Chamley, L. W. & James, J. L. Placental formation in early pregnancy: How is the centre of the placenta made?. *Hum. Reprod. Update* **24**, 750–760 (2018).
130. Wang, Y. & Zhao, S. *Vascular Biology of the Placenta. Integrated Systems Physiology: from Molecules to Function to Disease* (Morgan & Claypool Life Sciences, 2010).
131. Frosst, N. & Hinton, G. Distilling a neural network into a soft decision tree. arXiv preprint arXiv:1711.09784 (2017).
132. Tian, L., Wu, W. & Yu, T. Graph random forest: A graph embedded algorithm for identifying highly connected important features. *Biomolecules* **13**, 1153 (2023).
133. Wang, X. & Shen, H. W. GNNBoundary: Towards explaining graph neural networks through the lens of decision boundaries. In *The Twelfth International Conference on Learning Representations* (2024).
134. Naeini, M. P., Cooper, G. & Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29 (2015).
135. Yuan, M., Lang, B. & Quan, F. Student-friendly knowledge distillation. *Knowl.-Based Syst.* **296**, 111915 (2024).
136. Wu, F. et al. Simplifying graph convolutional networks. In *International Conference on Machine Learning*, 6861–6871 (PMLR, 2019).
137. Kotchoni, R. Detecting and measuring nonlinearity. *Econometrics* **6**, 37 (2018).


## Acknowledgements

The National Institutes of Health (NHLBI) supported a portion of the TB dataset preparation through grant R01HL14541 awarded to Dr. Gillian Beamer.

## Author contributions

B.Y. and V.A conceived the experiments. V.A. and B.Y. conducted the experiments. V.A., B.Y. and G.B analyzed the results. All authors reviewed the manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-13697-7.

**Correspondence** and requests for materials should be addressed to V.A.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.