IJEEC

International Journal of Electrical Engineering and Computing
Vol. 10, No. 1 (2026)

Original research paper
UDC
DOI

Edge-Aware Graph Neural Network Baselines for Protein Function Prediction on OGBN-Proteins

Aleksandar Stanković, Dejan Lisica
Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
E-mail address: stankovic.sv25.2022@uns.ac.rs, lisica.sv49.2022@uns.ac.rs

Abstract—**This paper presents an engineering-oriented study of edge-aware graph neural network baselines for protein function prediction on the OGBN-Proteins benchmark. The benchmark represents a large protein-protein interaction graph whose edges carry eight-dimensional association evidence and whose proteins have 112 functional labels. The study focuses on practical design choices that strongly affect reproducibility and deployment cost: construction of node features from edge evidence, use of scalar edge weights inside message passing, normalization under species-level distribution shift, and post-hoc decision calibration. We compare multilayer perceptrons, GraphSAGE, and GIN baselines in PyTorch Geometric, using mean, sum, and max edge-to-node aggregation, Batch Normalization, Layer Normalization, and a species-conditioned Layer Normalization variant. Results are reported over three seeds with ROC-AUC, micro-F1, calibrated micro-F1, expected calibration error, training time, memory use, and parameter count. Sum aggregation is consistently the strongest edge-to-node construction. GraphSAGE with sum-based features forms the best accuracy-cost trade-off, with Batch Normalization reaching the highest ROC-AUC and conditional Layer Normalization retaining stronger fixed-threshold behavior. Per-label temperature scaling and per-label thresholds substantially improve multi-label decision quality with negligible change in ROC-AUC, while light label-correlation smoothing yields small additional gains. The resulting protocol provides a compact, reproducible baseline for large edge-attributed biological graph settings. Together, these findings give practitioners clear default choices for feature construction, normalization, and decision calibration on edge-attributed protein graphs.**

Keywords—graph neural networks; protein function prediction; OGBN-Proteins; edge features; calibration; PyTorch Geometric

I. INTRODUCTION

Protein function prediction is an important task in computational biology because experimentally validated annotations are expensive, incomplete, and unevenly distributed across species. Protein-protein interaction networks offer a natural structure for this problem: proteins are represented as nodes and different kinds of experimental or database evidence are represented as edge-level association scores. A learning system that can combine graph topology with such edge evidence can help prioritize candidate functions for proteins that are poorly characterized.

The OGBN-Proteins benchmark from the Open Graph Benchmark (OGB) [1] is a standardized large-scale setting for this problem. It contains a protein-protein interaction graph with eight-dimensional edge attributes and 112 binary functional labels per protein. The official split is particularly challenging because validation and test nodes come from species not seen during training. As a result, strong performance requires not only graph modeling but also robust choices about normalization, edge feature use, and decision thresholds under cross-species distribution shift.

Graph neural networks (GNNs) are well suited to protein interaction graphs, but published benchmark numbers alone are often insufficient for practitioners. Small engineering choices, such as replacing mean aggregation with sum aggregation for incident edge evidence, may change both predictive quality and the memory or runtime required to obtain it. Reproducing a baseline inside a specific software stack can also be time-consuming, especially when edge features, calibration, and multi-label decision rules are handled differently across implementations.

This paper therefore studies edge-aware GNN baselines for OGBN-Proteins with a deliberately practical focus. We do not introduce a new architecture. Instead, we quantify how simple, reproducible choices shift the accuracy-cost frontier for a large edge-attributed biological graph. The study compares node features constructed from incident edge attributes, message-passing models that use scalarized edge evidence, normalization variants, multilayer perceptron (MLP) controls without graph propagation, and post-hoc calibration with label-wise thresholding.

The main contributions are as follows. First, we give a reproducible PyTorch-Geometric protocol for OGBN-Proteins that reports both predictive metrics and resource use. Second, we show that sum aggregation of edge evidence is a simple and strong default for this benchmark. Third, we compare Batch Normalization, Layer Normalization, and conditional Layer Normalization under the species split. Fourth, we show that calibrated per-label thresholds are essential when ROC-AUC is not the only metric of interest. Finally, we examine a light training-only label-correlation smoother as a low-cost way to exploit dependencies among functional labels.

This paper is a revised and expanded version of the paper presented at the XXV International Symposium INFOTEH-JAHORINA 2026

II. Background and Related Work

Let $G = (V, E)$ denote the protein interaction graph. Each edge (i, j) has an attribute vector $e_{ij} \in [0,1]^8$ representing association evidence, and each node i has a multi-label target $y_i \in \{0,1\}^{112}$. A model produces logits $\hat{z}_i \in R^{112}$, which are converted to probabilities with a sigmoid function. The official benchmark metric is the mean ROC-AUC over labels, while deployed multi-label systems also require thresholds to turn probabilities into binary decisions.

Message-passing neural networks provide the general template for GNNs [2]. At each layer, a node receives messages from neighboring nodes, aggregates them, and updates its representation. GraphSAGE [3] uses inductive neighbor aggregation, while GIN [4] emphasizes sum-like aggregation followed by a multilayer perceptron. Edge-conditioned and gated message passing methods use edge attributes to modulate messages, either through learned weights or edge-dependent gates [2], [5], [6].

Normalization is another important component. Batch Normalization (BN) [7] normalizes activations using batch statistics, while Layer Normalization (LN) [8] normalizes each example independently. Conditional normalization methods modulate affine parameters using auxiliary information [9]. In a species-split benchmark, species descriptors can serve as side information for a conditional Layer Normalization (CLN) variant, allowing the normalization layer to adapt without changing the message-passing backbone.

Finally, the benchmark is multi-label. ROC-AUC is threshold-free, but practical use requires calibrated probabilities and operating thresholds. Temperature scaling is a simple post-hoc calibration method [10], and related calibration methods include Platt scaling, isotonic regression, Bayesian binning, and Dirichlet calibration [11]-[14]. Multi-label prediction can also benefit from label dependency models such as classifier chains [15] or label-graph approaches [16]. In this work, we keep label dependency modeling deliberately lightweight to avoid masking the behavior of the GNN baselines themselves.

Edge attributes are especially important in biological graphs because an interaction edge is rarely just a binary fact. Association scores may summarize experiments, text-mining evidence, database annotations, co-expression, or other heterogeneous signals. Collapsing such evidence too early can remove useful information, but using a high-capacity edge network can also increase memory cost and make the baseline harder to reproduce. The present study therefore occupies a middle ground: it preserves the eight evidence channels in the node-feature construction and uses a transparent scalar edge strength in message passing.

This choice is also motivated by the role of baselines in graph machine learning. A baseline for a large biological graph should be easy to rerun, easy to audit, and strong enough that future architectures must improve on a meaningful reference point. For OGBN-Proteins, a weak baseline may lead to overstating the value of a new model, while an overly complex baseline can make it difficult to determine which design choice produced the gain. Separating edge-to-node aggregation, message weighting, normalization, and calibration makes the comparison more interpretable.

The species split adds another layer of difficulty. Random node splits can overestimate performance because nearby proteins and related interaction patterns may appear in both training and evaluation sets. A species split requires the model to transfer to proteins whose biological context differs from the training distribution. This makes OGBN-Proteins closer to practical annotation transfer, where a model trained on well-studied organisms is often used to support predictions in less completely characterized organisms.

The calibration literature is relevant because protein function prediction is often used as a ranking and prioritization tool rather than as a pure classification exercise. A laboratory user may ask for candidates above a confidence threshold, or for all functions that satisfy a recall or precision target. In those settings, uncalibrated probabilities can be misleading even if ROC-AUC is high. This is why the experiments report thresholded F1 and ECE in addition to AUC.

Label dependencies are similarly important. Protein functions are not independent: some functions co-occur because they belong to related biological processes, while others are mutually rare. However, learning a full label graph can add many degrees of freedom and may obscure the contribution of the node encoder. The training-only co-occurrence smoother used here is deliberately small. It tests whether label dependence is visible in the logits without turning the study into a new label-graph modeling paper.

III. Materials and Methods

A. Dataset and task

We follow the standard OGBN-Proteins task protocol [1]. The graph nodes are proteins, the edges encode eight channels of protein association evidence, and each protein is associated with 112 binary labels. The split is species based: training uses proteins from training species, validation uses an unseen species, and testing uses another unseen species. In the available split used for these experiments, the validation species is mouse (NCBI 10090) and the test species is zebrafish (NCBI 7955).

The input graph does not provide ordinary node attributes. We therefore construct initial node features from incident edge attributes. For a node i, the multiset of adjacent edge vectors is aggregated channel-wise with one of three functions: mean, sum, or max. Mean aggregation measures average association evidence, sum aggregation preserves both evidence strength and interaction volume, and max aggregation keeps the strongest evidence channel observed around the node. This stage is isolated in the experiments so that its effect can be measured independently of the GNN architecture.

B. Models

The no-graph control is an MLP trained directly on the edge-to-node features. It predicts 112 logits with binary cross-entropy loss and uses the same train, validation, and test splits as the GNNs. MLP variants differ in depth, hidden width, and normalization. This control answers an important question: how much of the benchmark signal is already captured by local edge evidence before any message passing is applied?

For graph baselines, we evaluate GraphSAGE and GIN. The principal message-passing experiments use three layers and hidden size 512. Edge vectors are scalarized into non-negative edge strengths and used as message weights. The default scalarizer is the channel sum, while a learned one-dimensional scalarizer is also evaluated. This design keeps the models simple and reproducible while still allowing edge attributes to influence

both the initial node representation and the messages exchanged over the graph.

Normalization variants are compared with the same backbone whenever possible. BN uses batch statistics in hidden layers, LN normalizes each node representation independently, and CLN modulates the affine parameters of Layer Normalization using a species descriptor. The descriptor is computed without using labels and contains simple species-level summary statistics: log species size, mean node degree, mean incident edge-evidence sum, and standard deviation of incident edge-evidence sum. These descriptor values are standardized using training-species statistics.

For hidden representation h_i of node i and its species descriptor s_i, CLN first applies ordinary Layer Normalization and then uses a small conditioning network to predict affine offsets. In particular, the conditioning network is a two-layer MLP that maps s_i to $\Delta\gamma(s_i)$ and $\Delta\beta(s_i)$. The normalized representation is then computed as $CLN(h_i, s_i) = (1 + \Delta\gamma(s_i)) \odot LN(h_i) + \Delta\beta(s_i)$. The conditioning network is shared across all nodes and is applied only inside the normalization layers. It does not modify the GraphSAGE or GIN message-passing operators. CLN is therefore intentionally lightweight as it tests whether species-aware normalization can improve robustness on the official species split without introducing a new graph convolution architecture.

C. Training, evaluation, and post-hoc analysis

All models are implemented in PyTorch Geometric [17]. Runs use seeds 1, 2, and 3, early stopping on validation ROC-AUC, and binary cross-entropy with logits. Unless stated otherwise, performance metrics are reported as mean ± standard deviation over the three seeds. Because several differences between model variants are small, we interpret such differences as numerical trends rather than statistically definitive improvements when their ranges overlap. For each run we record validation and test ROC-AUC, fixed-threshold micro-F1 at probability threshold 0.5, parameter count, wall-clock training time, and peak GPU memory. The experiments were executed on an NVIDIA A800-SXM4-40GB GPU. Reporting resource use is important because the best benchmark number is not necessarily the best engineering choice for a deployable pipeline.

For post-hoc calibration, validation logits are used to fit either a global temperature or per-label temperatures regularized toward a global value. Temperatures are optimized by minimizing validation negative log-likelihood. After calibration, a separate threshold is selected for each label by maximizing validation F1, with a fallback threshold based on ROC behavior when the validation labels are degenerate. Test results are reported without refitting. Calibration quality is measured with expected calibration error (ECE) and the Brier score [18], in addition to AUC and F1.

To test label dependency without introducing a large additional model, we derive a label co-occurrence matrix P from training labels only. Rows of P are normalized so that P_{jk} reflects how often label k co-occurs with label j in the training set. Test-time smoothing is applied in logit space as $z' = z + \lambda z P^T$, with a small λ chosen on validation data. Because P is computed only from training labels, the procedure avoids leakage from validation or test labels.

D. Reproducibility safeguards

Several safeguards are used to keep the protocol reproducible and leakage-free. All transformations that depend on labels, including threshold selection and label co-occurrence estimation, are fitted only on the training or validation partition specified for that stage. Test labels are used only for final metric calculation. The same random seeds are reused across related configurations so that differences are more likely to reflect modeling choices than incidental training variation.

The reported ROC-AUC is the mean over the 112 labels for which the metric is defined. Fixed-threshold micro-F1 is computed by applying probability threshold 0.5 to every label. Calibrated micro-F1 is computed after applying the validation-fitted temperatures and thresholds. ECE is computed by binning predicted probabilities and comparing average confidence to empirical frequency. Although ECE is an imperfect summary for multi-label data, it gives a useful indication of whether predicted probabilities are suitable for threshold-based decisions.

Resource measurements are reported because the benchmark graph is large enough that engineering constraints matter. Parameter count describes model size, GPU memory describes deployability on available hardware, and wall-clock time describes the cost of repeated experimentation. A model with marginally higher AUC but much higher memory use may be less attractive than a simpler model for routine reruns, ablation studies, or downstream biological workflows.

The implementation, configuration files, and analysis scripts are publicly available at https://github.com/SV25-22/ECHO-Proteins. The repository is intended to support independent checking of the reported protocol, including edge-to-node feature construction, model training, calibration, threshold selection, and label-correlation smoothing.

E. Additional protocol details

The aggregation stage is intentionally separated from the neural encoder. This separation makes the input representation deterministic and allows the MLP and GNN experiments to start from exactly the same node features. It also helps diagnose whether a gain comes from message passing or from a better summary of edge evidence. In practical systems, this distinction is useful because edge-to-node aggregation can be cached once and reused across many model variants.

The scalar edge weights used inside the GNN are treated as message strengths. With the default sum scalarizer, an edge with stronger total association evidence contributes more to the aggregated message. The learned scalarizer is a small one-dimensional mapping from the eight edge channels to a scalar strength. It tests whether the model benefits from learning channel importance, while keeping the parameter increase minimal compared with a full edge-conditioned convolution.

The CLN variant uses side information only in the normalization affine parameters. This design is deliberately less intrusive than adding species embeddings directly to node features or changing the graph convolution. It allows the message-passing backbone to remain comparable with the BN and LN variants while still testing whether species information can help the hidden representation adjust to the benchmark split.

Early stopping is based on validation AUC because AUC is the official benchmark metric and is less sensitive to a particular decision threshold. However, the selected checkpoint is later evaluated with both threshold-free and thresholded metrics. This mirrors a common workflow in which model selection follows the benchmark objective, while deployment analysis checks whether the chosen model produces usable decisions.

The threshold search is performed independently for each label because label frequencies vary substantially. A single global threshold can be dominated by common labels and can suppress rare labels even when their ranking quality is acceptable. Per-label thresholds are therefore a simple way to adapt the decision rule to label imbalance without retraining the neural network or changing the loss function.

F. Statistical interpretation protocol

Because the study is intended as an engineering baseline rather than a new state-of-the-art architecture, the statistical analysis focuses on stability and practical effect size. For each repeated configuration, we summarize the observed variation across seeds with the standard deviation. We do not treat small differences as conclusive when their seed-level ranges overlap. Instead, such cases are described as comparable, numerically close, or as trends that should be verified by additional repetitions.

This interpretation is important for OGBN-Proteins because several reported differences are small relative to the absolute metric values. A difference of a few thousandths in ROC-AUC may not be practically meaningful if it is similar to the run-to-run variation caused by initialization, minibatch ordering, or early stopping. For this reason, the main conclusions emphasize repeated patterns across related experiments rather than a single best number. For example, the recommendation to use sum edge-to-node aggregation is supported not only by one GraphSAGE comparison, but also by the MLP controls and the aggregation ablation.

For thresholded metrics, we interpret changes more cautiously because micro-F1 depends on both the learned score distribution and the chosen decision threshold. Fixed-threshold F1 is useful as a diagnostic of raw probability behavior, while calibrated F1 is interpreted as the performance of the complete decision pipeline. This distinction prevents calibration gains from being mistaken for architectural gains in the encoder.

IV. RESULTS

A. Main graph baselines

Table I summarizes the main GNN baselines under the standardized three-layer, hidden-size-512 setup with sum edge-to-node features. GraphSAGE forms the strongest accuracy-cost region in this comparison. SAGE with BN reaches the highest validation AUC and ties the best test AUC, while SAGE with CLN reaches the same test AUC and substantially higher fixed-threshold micro-F1 at 0.5. Because the test AUC values of BN and CLN overlap within seed-level variation, we interpret their ranking performance as comparable rather than statistically separable. LN is close in AUC and remains competitive, especially when combined with sum aggregation.

The GIN baseline is weaker in ROC-AUC than GraphSAGE, although its fixed-threshold F1 is higher than SAGE+BN. This contrast illustrates a central theme of the benchmark: ranking metrics and decision metrics can tell different stories. A model may rank positive labels well enough to achieve high AUC while producing probabilities whose default threshold behavior is poor.

TABLE I. MAIN GNN BASELINES ON OGBN-PROTEINS (3 LAYERS, HIDDEN SIZE 512, SUM EDGE-TO-NODE FEATURES; MEAN ± STANDARD DEVIATION OVER 3 SEEDS).

Model	Norm / scalar	Val AUC	Test AUC	Test F1	Params (M)
SAGE	BN / sum	0.864 ± 0.003	0.792 ± 0.004	0.096 ± 0.011	0.65
SAGE	CLN / sum	0.855 ± 0.004	0.792 ± 0.005	0.145 ± 0.009	1.71
SAGE	LN / learned1d	0.850 ± 0.005	0.787 ± 0.004	0.128 ± 0.010	0.65
SAGE	LN / sum	0.855 ± 0.004	0.789 ± 0.004	0.144 ± 0.008	0.65
GIN	BN / sum	0.845 ± 0.006	0.761 ± 0.006	0.149 ± 0.012	0.85

B. MLP controls

Table II reports representative MLP baselines. These models do not use graph message passing and therefore measure the strength of edge-aggregated node features alone. Sum aggregation dominates the top MLP configurations, and the best MLPs achieve test AUC around 0.743. This is lower than the best GraphSAGE models but strong enough to show that local edge evidence is highly informative.

The MLP results also explain why edge-to-node feature construction deserves careful treatment. Mean aggregation loses useful information about interaction volume, while max aggregation keeps only the strongest channel-wise evidence. Sum aggregation preserves more of the total association signal and repeatedly appears in the best-performing configurations. Batch Normalization improves the strong sum-based MLPs, whereas no-normalization variants are less stable.

TABLE II. REPRESENTATIVE MLP BASELINES. PERFORMANCE METRICS ARE REPORTED AS MEAN ± STANDARD DEVIATION OVER THREE SEEDS. VRAM IS IN MB AND PARAMS IN MILLIONS.

Config	Val AUC	Test AUC	Test F1	VRAM	Params (M)
sum_deep_none	0.796 ± 0.004	0.743 ± 0.005	0.145 ± 0.012	847	0.18
sum_taper_bn	0.799 ± 0.003	0.743 ± 0.004	0.128 ± 0.010	1075	0.19
sum_deep_bn	0.802 ± 0.003	0.742 ± 0.004	0.125 ± 0.009	1073	0.18
sum_uniform_256_bn	0.795 ± 0.004	0.740 ± 0.005	0.137 ± 0.011	732	0.10
max_uniform_256_cln_desc	0.741 ± 0.006	0.715 ± 0.006	0.120 ± 0.010	1021	0.10
mean_uniform_256_none	0.636 ± 0.008	0.577 ± 0.009	0.075 ± 0.008	603	0.10
sum_taper_none	0.447 ± 0.012	0.465 ± 0.011	0.049 ± 0.006	848	0.18

C. Edge-to-node aggregation

Table III isolates the edge-to-node aggregation choice for SAGE with Layer Normalization. Replacing mean with sum improves both validation and test AUC. Max aggregation is competitive, but it remains slightly below sum aggregation. The learned one-dimensional edge scalarizer is also close to sum, but it does not consistently surpass the simple channel-sum scalarizer.

TABLE III. EDGE-TO-NODE AGGREGATION ABLATION FOR SAGE WITH LAYERNORM (VALUES ARE REPORTED AS MEAN ± STANDARD DEVIATION OVER THREE SEEDS).

Variant	Agg / scalar	Val AUC	Test AUC	Test F1	Params (M)
SAGE (LN)	max / sum	0.824 ± 0.006	0.779 ± 0.005	0.141 ± 0.010	0.65
SAGE (LN)	mean / sum	0.839 ± 0.005	0.775 ± 0.006	0.138 ± 0.011	0.65
SAGE (LN)	sum / learned1d	0.850 ± 0.005	0.787 ± 0.004	0.128 ± 0.010	0.65
SAGE (LN)	sum / sum	0.855 ± 0.004	0.789 ± 0.004	0.144 ± 0.008	0.65

These results support a conservative engineering recommendation. Before adding more complex edge networks, practitioners should test the simple sum-based construction. It is cheap, deterministic, easy to reproduce, and strong across both the MLP controls and the GraphSAGE baselines. More expressive edge functions may still be useful, but they should be compared against this baseline rather than against a weaker mean-only setup. Because the differences among the top aggregation variants are small, we report this ablation numerically in Table III rather than duplicating it as a bar plot.

D. Species-level transfer

The species split is central to the benchmark. The validation split contains mouse proteins and the test split contains zebrafish proteins, so the evaluation measures cross-species transfer rather than random held-out nodes. Fig. 1 compares per-species AUC for the normalization variants. BN and CLN are close within seed-level variability on the validation species, and both remain strong on the test species. LN is slightly behind but still competitive.

CLN is not a universal replacement for BN in these experiments. Instead, it should be interpreted as a useful alternative when species-aware conditioning is desired. It keeps ranking performance near the AUC frontier and has stronger fixed-threshold behavior than BN. However, because the validation and test partitions each contain one species in this setup, we avoid drawing broad biological conclusions from this single species pair.

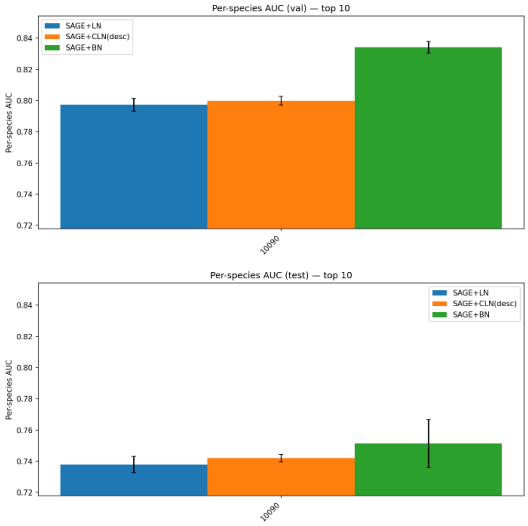

Figure 1. Per-species AUC for SAGE normalization variants, reported as mean ± standard deviation over three seeds. Top: validation species (mouse, NCBI 10090). Bottom: test species (zebrafish, NCBI 7955).

TABLE IV. TRAINING-DERIVED LABEL CO-OCCURRENCE MATRIX SUMMARY.

Statistic	Value	Meaning
Number of labels (K)	112	Total functional labels
Mean off-diagonal P_{jk}	0.009	Average conditional co-occurrence mass
Max off-diagonal P_{jk}	0.061	Strongest conditional relation
Min off-diagonal P_{jk}	0.0003	Weakest conditional relation

E. Efficiency, calibration, and label smoothing

The accuracy-cost plot (Fig. 2) shows that the strongest practical region is occupied by SAGE variants. The best BN and CLN configurations have similar AUC, while CLN uses more parameters because it conditions normalization parameters on species descriptors. In contrast, GIN does not compensate for its additional cost with higher AUC in this setting. The MLP controls are cheaper, but their lower AUC shows the value of graph propagation once a strong edge-to-node representation has been built.

From an engineering standpoint, the frontier matters more than a single best number. If the goal is maximum AUC and batch statistics are acceptable, SAGE+BN is the most direct choice. If calibrated or thresholded outputs are important, SAGE+CLN or SAGE+LN may be easier to deploy after temperature scaling. If hardware is limited or many exploratory runs are needed, the sum-based MLP can serve as a fast diagnostic before committing to full graph training.

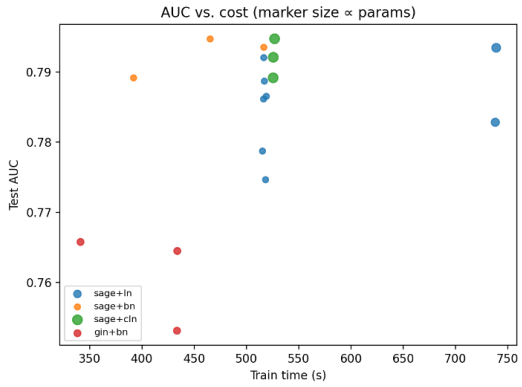

Figure 2. AUC versus training cost. Marker size is proportional to parameter count; color encodes model and normalization.

Calibration and thresholding change the practical interpretation of the models. The large improvement in calibrated micro-F1 should not be interpreted as a change in the ranking ability of the encoder. ROC-AUC is threshold-free and depends mainly on the relative ordering of positive and negative examples for each label. Temperature scaling is monotonic for each label and therefore leaves this ordering nearly unchanged. In contrast, micro-F1 depends directly on the binary decisions produced after thresholding. The default threshold of 0.5 is poorly matched to a highly imbalanced multi-label problem, where different labels have different prevalences and score distributions. Per-label threshold selection adjusts the operating point of each label using validation data, which explains why thresholded F1 can improve substantially even when ROC-AUC remains almost constant. For this reason, calibration and threshold selection should be treated as part of the decision pipeline rather than as an architectural improvement.

Light label-correlation smoothing gives small additional improvements for LN and CLN and slightly reduces ECE. BN remains less well calibrated even after post-hoc correction. This suggests that reporting only ROC-AUC is insufficient for multi-label protein function prediction. A deployment-oriented evaluation should include at least one thresholded metric and a calibration metric.

TABLE V. TEST METRICS BEFORE AND AFTER LABEL-CORRELATION SMOOTHING.

Model	AUC	micro-F1 cal+thr	ECE
SAGE+LN	*0.792 → 0.794 (+0.002)*	*0.795 → 0.796 (+0.001)*	*0.188 → 0.178 (−0.010)*
SAGE+CLN	*0.795 → 0.796 (+0.001)*	*0.786 → 0.787 (+0.001)*	*0.183 → 0.173 (−0.010)*
SAGE+BN	*0.795 → 0.794 (−0.001)*	*0.592 → 0.587 (−0.005)*	*0.350 → 0.348 (−0.002)*

Each cell reports the metric before → after label-correlation smoothing, with the absolute change (after − before) in parentheses. Higher is better for AUC and micro-F1; for ECE, lower is better, so a negative change indicates improvement. All variants use sum edge aggregation.

The smoothing results should therefore be read as a diagnostic rather than as evidence that a label-correlation model is sufficient for protein function prediction. The co-occurrence matrix is computed only from training labels and is intentionally low capacity. It can slightly adjust logits toward commonly co-occurring functions, but it cannot model the full structure of biological function annotations. In particular, it does not encode the Gene Ontology hierarchy, parent-child relations, or evidence-code differences. This design keeps the post-hoc analysis leakage-free and easy to reproduce, but it also limits the expected size of the improvement.

The practical value of this experiment is that it separates three effects that are often mixed together: ranking quality from the neural encoder, decision quality from calibration and thresholds, and dependency adjustment from label smoothing. The results suggest that the largest gain comes from the decision stage, while label-correlation smoothing provides only small additional changes. This supports the use of label smoothing as a lightweight optional step rather than as a replacement for stronger label-graph models. It also suggests that future work should evaluate learned label dependencies as a separate component with matched validation rules, rather than attributing all post-hoc improvements to the GNN backbone.

The gap between MLP and GraphSAGE performance clarifies the value of graph propagation. The MLP sees only the aggregated evidence incident to each node, so it can exploit local interaction strength but not the labels or evidence patterns of neighboring proteins. GraphSAGE improves over this control by propagating information through the interaction graph. The gain is not overwhelming, which means the local evidence is already strong, but it is consistent enough to justify message passing when resources allow it.

The learned edge scalarizer does not clearly outperform the channel-sum scalarizer. One possible interpretation is that the eight evidence channels are already calibrated enough that total evidence is a good proxy for message strength. Another possibility is that the model and validation split do not provide enough signal for a very small scalarizer to learn a better weighting. In either case, the result is useful: it establishes that a deterministic scalarizer should be included as a serious baseline.

The difference between fixed-threshold F1 and calibrated F1 is larger than the differences among several model backbones. This does not reduce the importance of architecture choices, but it changes how results should be interpreted. A model that is slightly weaker in AUC can still be preferable if its outputs are easier to calibrate and threshold. For users who need binary predictions, calibration is part of the model pipeline rather than a cosmetic post-processing step.

The label-correlation smoother is intentionally weak, and its gains are therefore expected to be modest. Stronger smoothing could improve some labels but might also propagate errors between unrelated functions. The small value of the largest off-diagonal co-occurrence entry (Table IV) supports this caution: most label pairs share little conditional mass. A light smoother is appropriate for a baseline, while a learned label graph should be evaluated as a separate model component.

The results also show why multiple metrics are necessary. AUC, fixed-threshold F1, calibrated F1, ECE, memory, and parameter count each capture a different failure mode. A model can rank labels well but be poorly calibrated; it can be well calibrated but too expensive to rerun; or it can be fast but too weak for the final task. Reporting the full set of metrics makes these trade-offs visible.

V. DISCUSSION

The experiments show that a careful baseline can be more informative than a more elaborate model whose engineering assumptions are unclear. Sum aggregation is a particularly

strong example. It is a simple operation, but on OGBN-Proteins it captures useful information about the amount of association evidence incident to a protein. This makes it a better default than mean aggregation for the configurations evaluated here.

GraphSAGE forms the best accuracy-cost frontier in this study. It improves substantially over the MLP controls while keeping parameter counts and memory use modest. GIN remains a useful comparison, but it does not match the GraphSAGE AUC. The results do not imply that GIN is generally unsuitable for biological graphs; rather, they show that under this edge-aware preprocessing and training protocol, GraphSAGE is the stronger baseline to beat.

The normalization results are more nuanced. BN gives the highest AUC, but its default threshold behavior is weak. CLN trades additional parameters for stronger fixed-threshold behavior and competitive AUC. LN is slightly weaker in ranking quality but easier to reason about under distribution shift because it does not depend on batch statistics. The right choice therefore depends on whether the system is optimized for leaderboard AUC, immediate thresholded decisions, or robustness and simplicity.

The calibration results are the clearest practical lesson. In multi-label protein function prediction, ROC-AUC measures ranking but not the quality of binary decisions. Per-label temperature scaling and per-label thresholds are lightweight, post-hoc, and highly effective. They should be included whenever model outputs are intended to support decisions such as candidate function screening, follow-up experiments, or label prioritization.

Table VI distills the experimental findings into practical recommendations. These recommendations are intentionally conservative. They are meant as a starting point for reproducible experimentation rather than as universal rules for every protein graph. Their value is that each recommendation follows from a small, inspectable design choice that can be implemented without specialized architecture code.

TABLE VI. PRACTICAL RECOMMENDATIONS DERIVED FROM THE ABLATIONS.

Finding	Recommendation
Sum edge-to-node features dominate mean and max in the evaluated settings.	Use sum aggregation as the first baseline for edge-attributed OGBN-Proteins experiments.
GraphSAGE gives the strongest AUC-cost trade-off.	Treat SAGE with sum features as the reference model before testing heavier architectures.
BN reaches high AUC but weak fixed-threshold F1.	Pair BN models with post-hoc calibration and threshold selection before deployment.
CLN is competitive in AUC and stronger at default thresholds.	Use CLN when species-aware conditioning is useful and extra parameters are acceptable.
Calibrated thresholds dominate the default 0.5 rule.	Report calibrated micro-F1 and ECE alongside ROC-AUC for multi-label decisions.

A practical implication of this staged protocol is that failures become easier to diagnose. If the MLP control performs poorly, the likely issue is in preprocessing, label alignment, or edge-to-node aggregation. If the MLP is strong but the GNN is weak, the issue may lie in message weighting, oversmoothing, or normalization. If ROC-AUC is strong but fixed-threshold F1 is weak, the most likely cause is not necessarily the graph encoder, but the calibration and thresholding stage. This separation is useful for reproducibility because it allows researchers to inspect each component of the pipeline independently.

The same separation is also useful for reporting. AUC values alone are not enough to diagnose practical failure modes in a multi-label biological task. Whenever possible, saved logits, labels, node identifiers, and species identifiers should be retained so that calibration, thresholding, ECE, and label-smoothing analyses can be recomputed without retraining the model. This makes revision and independent checking easier, especially when reviewers request additional thresholding rules or calibration summaries.

The study also has limitations. The experiments use a single benchmark and a specific species split, so conclusions should be checked on additional protein-interaction datasets. The CLN implementation uses species descriptors only in the normalization layer; richer domain adaptation methods may perform better. The label-correlation smoother is intentionally simple and cannot represent the full Gene Ontology structure or asymmetric biological dependencies. Finally, the edge-aware message passing uses scalarized edge weights rather than a fully learned edge network, which leaves room for stronger but more expensive architectures.

Future work should evaluate learned edge-conditioned message passing at matched parameter and memory budgets, graph transformers with edge-derived attention biases [19], [20], stronger calibration methods, and learned label-graph models that incorporate Gene Ontology priors while preserving strict train-validation-test separation.

VI. CONCLUSION

This paper presented a reproducible edge-aware baseline study for protein function prediction on OGBN-Proteins. The results show that sum aggregation of incident edge evidence is a strong node-feature construction, GraphSAGE is a robust and efficient message-passing backbone, and normalization choices affect ranking and thresholded behavior differently. BN reaches the highest ROC-AUC, while CLN remains close in AUC and improves fixed-threshold behavior.

The most important deployment lesson is that post-hoc calibration and per-label thresholds are not optional details. They change micro-F1 substantially while leaving ROC-AUC almost unchanged. Light label-correlation smoothing adds small gains when computed only from training labels. Together, these findings provide a practical baseline protocol for large edge-attributed biological graphs and a clear reference point for future edge-aware GNN architectures.

ACKNOWLEDGMENT

We gratefully acknowledge computational support from Xinming Wang at the Institute of Automation, Chinese Academy of Sciences (CASIA).

REFERENCES

[1] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open Graph Benchmark: Datasets for machine learning on graphs," in Advances in Neural Information Processing Systems, 2020. URL: https://ogb.stanford.edu/

[2] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in ICML, 2017. URL: https://arxiv.org/abs/1704.01212

[3] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in NeurIPS, 2017. URL: https://arxiv.org/abs/1706.02216

[4] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in ICLR, 2019. URL: https://arxiv.org/abs/1810.00826

[5] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in ICLR, 2016. URL: https://arxiv.org/abs/1511.05493

[6] X. Bresson and T. Laurent, "Residual gated graph ConvNets," arXiv:1711.07553, 2017. URL: https://arxiv.org/abs/1711.07553

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in ICML, 2015. URL: https://arxiv.org/abs/1502.03167

[8] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv:1607.06450, 2016. URL: https://arxiv.org/abs/1607.06450

[9] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in AAAI, 2018. URL: https://arxiv.org/abs/1709.07871

[10] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in ICML, 2017. URL: https://arxiv.org/abs/1706.04599

[11] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers, 1999.

[12] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in KDD, 2002.

[13] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well-calibrated probabilities using Bayesian binning into quantiles," in AAAI, 2015.

[14] M. Kull, M. Perello-Nieto, M. Kangsepp, T. Silva Filho, H. Song, and P. Flach, "Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration," in NeurIPS, 2019. URL: https://arxiv.org/abs/1910.12656

[15] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in ECML PKDD, 2009.

[16] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in CVPR, 2019. URL: https://arxiv.org/abs/1904.03582

[17] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in ICLR Workshop, 2019. URL: https://arxiv.org/abs/1903.02428

[18] G. W. Brier, "Verification of forecasts expressed in terms of probability," Monthly Weather Review, vol. 78, no. 1, pp. 1-3, 1950.

[19] C. Ying et al., "Do transformers really perform bad for graph representation?" in NeurIPS, 2021. URL: https://arxiv.org/abs/2106.05234

[20] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," in AAAI Workshop on Deep Learning on Graphs, 2021. URL: https://arxiv.org/abs/2012.09699

Aleksandar Stanković is a fourth-year student of Software Engineering and Information Technologies at the Faculty of Technical Sciences, University of Novi Sad. His research interests include graph machine learning, machine learning systems, GPU-accelerated graph analytics, temporal graphs, and social computing. In 2025, he was a research intern at the DataNet Lab at Fudan University in Shanghai, where he continues to collaborate as an external associate. He is the author of several research papers in the areas of graph analytics, temporal motifs, and efficient systems for graph machine learning. He is the recipient of the Annual Award of the Mathematical Institute of the Serbian Academy of Sciences and Arts in computer science for the best original student paper.

Dejan Lisica is a software engineering and information technology student at the Faculty of Technical Sciences, University of Novi Sad. His interests lie at the intersection of computer science and the life sciences, specifically focusing on machine learning, graph neural networks, and AI applications in drug discovery and protein design. He has developed projects involving molecular machine learning, protein function prediction, and graph-based modeling. Notably, his team achieved a high-ranking finish in the global DREAM Target 2035 Challenge, with their work featured in a poster presentation at the Mainframe Symposium. A recipient of the Fund for Young Talents of the Republic of Serbia scholarship, his current research focuses on applying modern ML methods to complex biological and chemical data.