Journal of Machine Learning Research 1–13

Genen: Refining Gene Representation and Mitigating Bias with Integrated LLMs and GNNs

Yue Hu[1], **David Arredondo**[1], **Kushal Virupakshappa**[1], **Oladimeji Macaulay**[1], **Luis Tafoya**[1], **Ala Jararweh**[1,2], **Yanfu Zhang**[3], **Avinash Sahu**[1,2]
YZHANG105@WM.EDU
ASAHU@SALUD.UNM.EDU
[1] *Comprehensive Cancer Center, The University of New Mexico,* [2] *Department of Computer Science, The University of New Mexico,* [3] *Department of Computer Science, William and Mary*

Abstract

Understanding how genes and proteins drive biological processes is fundamental for elucidating their implications for health and disease. While representation learning advances these understandings, it often overlooks the relational information inherent in gene networks and fails to address fairness and ethical concerns arising from data bias. Here, we present *Genen*, a model that integrates large language models (LLMs) with graph neural networks (GNNs) to refine gene representations by combining textual data with relational information propagation. *Genen* dynamically updates gene embeddings based on context, ensuring the model remains accurate as new genetic insights are discovered. Benchmarking *Genen* across eight diverse tasks revealed that its information propagation performs comparably to, and often surpasses, state-of-the-art methods. In solubility and localization tasks, we demonstrated *Genen's* robust inference mechanism in settings with incomplete data, achieving accurate predictions even with 75% of the data hidden. This suggests that GNNs are effective tools for handling incomplete datasets and mitigating biases by ensuring that genes with sparse data benefit from the knowledge of well-studied genes. *Genen's* integration of relational information with LLMs' textual understanding provides a promising approach for addressing biases and promoting equitable AI applications in genomic research. *Genen* code is publicly available at https://github.com/yuehu99/Genen.

Keywords: Gene properties, Information propogation, Graph Neural Network, LLM

1. Introduction

Understanding gene functions is fundamental to unraveling the complexities of biological processes and their implications for human health and disease. Genes encode proteins that drive critical biological processes, and their functions help explain the roles of these proteins in individual cells and overall human health. Despite the complexity and variability across different cellular, individual, and environmental contexts, advances in genetic research have allowed certain genes to be very well characterized. While traditional laboratory-based models capture only a small subset of these contexts, a large body of relational information between genes creates a large network of knowledge from which further insight can be inferred. Computational models have been developed to predict gene attributes, leveraging large datasets to enhance our understanding of gene functions.

The advent of foundation models has introduced a new paradigm in machine learning, offering versatile alternatives to traditional task-specific models. These models, once

pre-trained on extensive datasets, can be fine-tuned for a wide array of predictive tasks, often outperforming task-specific models [1]. Large Language Models (LLMs), for instance, utilize large text corpora to identify statistical relationships between words, encapsulating comprehensive knowledge in unstructured text. GeneLLM, a transformer-based model integrating textual information through contrastive learning, marked a step forward by leveraging unstructured text to enhance gene representation and achieve robust zero-shot learning capabilities.

However, existing methods, including GeneLLM, fail to capture the rich relational information inherent in gene interaction networks and gene ontologies, treating data in a linear fashion. This limitation reduces the biological relevance and predictive power of gene representations. Furthermore, AI models often inherit and amplify biases from training data, focusing on well-studied genes and neglecting those related to minority populations or less-tied to diseases. Traditional models also struggle with incomplete gene information, leading to significant performance degradation. Additionally, current models integrate textual data statically, which hampers their ability to adapt to new contexts, and many do not address fairness and ethical AI deployment in genomics, potentially perpetuating existing biases and disparities.

To comprehensively address these critical gaps, we present *Genen*, which enhances gene representation learning by integrating large language models (LLMs) with graph neural networks (GNNs). By integrating alternative contrastive learning that exploits the inherent graph structure of gene interaction data and gene ontologies, *Genen* enables gene embeddings to be updated dynamically and context-sensitively. Specifically, GNNs are able to capture and propagate complex relational information in gene networks, which is critical for accurately modeling gene interactions and functional annotations that are often overlooked by traditional models. In addition, *Genen* effectively addresses the challenge of missing information in genetic datasets and reduces bias. By leveraging the structural and relational capabilities of GNNs, *Genen* is able to infer missing gene properties through connections learned in gene interaction networks. Because the model can interpolate missing information based on the relational context provided by well-represented genes, the model maintains high performance even when data is incomplete, demonstrating the ability of GNNs to dynamically infer missing information. The integration of *Genen* allows gene embeddings to be updated context-dependently, keeping the model current and accurate. Committed to ethical AI deployment, *Genen* addresses training data bias and enhances the interpretability of predictions, setting a new standard for fair and responsible AI applications in genomic research.

2. Related Work

Gene and protein representation learnings have primarily focused on expression, sequence, or network data [21, 7, 13], driving advancements in gene-gene interaction and 3D structure predictions as well as cell property elucidation from single-cell RNA-Seq data [4]. Despite their efficacy in disease association and cancer classification, they primarily rely on quantitative data, potentially overlooking the contextual information embedded in textual sources. To improve protein representations, ProteinBERT [1] and OntoProtein [30] demonstrate the potential of integrating protein sequences with Gene Ontology (GO) using self-supervised

and contrastive learning, respectively. Our work extends this multimodal approach by incorporating GO annotations and textual information.

Recent advances in NLP models, such as BERT [5], LLaMA [23], and GPT [17], have revolutionized the utilization of unstructured biomedical texts from repositories like PubMed[1] and Europe PMC[2]. GeneLLM [11] builds on BioMedBERT [10], a BERT model further pre-trained on PubMed abstracts, by using contrastive learning to inject knowledge of GO summaries and their relationships into the model [22, 30]. However, the application of such models for gene and cell-specific predictions remains understudied.

Recent studies have sounded the alarm on the issue of bias amplification in AI [9], particularly in healthcare where biases in training data, such as knowledge biases, can result in significant disparities like the under-diagnosis of underserved populations [19]. These studies call for transparency and interpretability to ensure equitable healthcare [25]. Our work focuses on knowledge bias mitigation in gene data by using GNNs.

3. Methods

The *Genen* framework integrates a pretrained LLM with Graph GNN (Figure 1). This approach aims to directly update gene embedding based on the relationships between Gene Ontology and gene summaries. Here, we detail the sources of our data, followed by the configuration of the pretrained encoder, and then describe the parameter selection on various GNNs. Lastly, we discuss the graph structures used in *Genen*.

3.1. Data Collection and Preprocessing

The gene embeddings used in our study were generated from two publicly accessible databases. Gene descriptions were retrieved from MyGene [3] [28], and gene functions were obtained from UniProt. These two summaries were concatenated to form a comprehensive gene summary for each gene. During preprocessing, non-essential identifiers such as PubMed IDs, author names, and isoform IDs were removed, as well as duplicate gene summaries. After preprocessing, the dataset comprised a total of 14,450 unique gene summaries, which were then used to generate gene embeddings. The protein protein interaction (PPI) links were downloaded from the STRING database (https://string-db.org). Two genes were defined as interacting if any of their protein products interacted with each other. Duplicate links and self-loops were removed, resulting in 13,067,420 graph edges. Gene Ontology (GO) terms were obtained from the Gene Ontology website (http://geneontology.org). The data was processed using

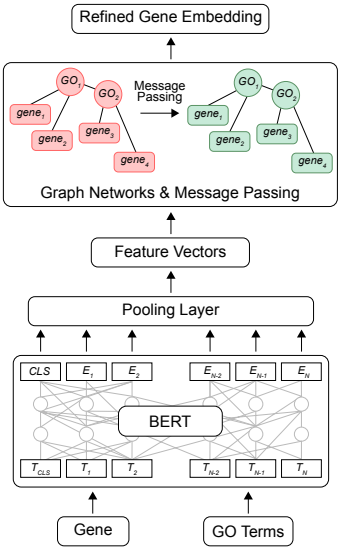

Figure 1 *Genen* Architecture . *Genen* integrates LLM and GNN. BERT encodes genes and GO terms into feature vectors, which are pooled and used to initialize nodes of GO networks. Message passing between gene and GO nodes refines gene embeddings dynamically to output context-sensitive gene representations.

1. https://pubmed.ncbi.nlm.nih.gov/
2. https://europepmc.org
3. https://mygene.info

the `obonet` Python package, resulting in 47,595 GO terms and their description, 83,796 GO-GO relationships, and 393,231 gene-GO relationships. The annotated solubility dataset from Dallago et al. [3] contains 1499 protein sets labeled *soluble* or *membrane*. Each gene is then labeled *soluble* if one of its protein products is labeled such in the solubility dataset, and *membrane* otherwise.

3.2. Large Language Encoder for Initial Embeddings

To obtain embeddings from text, we use GeneLLM-Base [11], which is the non-finetuned GeneLLM or BioMedBERT [10] with pooling. To generate initial (GeneLLM-Base) embeddings, each gene or GO term text summary is tokenized into a sequence of N tokens $(x_1, \ldots, x_N)$, including special tokens [CLS] and [SEP]. These tokens are input into GeneLLM-Base, which then outputs a series of token embeddings $(T_1, \ldots, T_N)$. GeneLLM-Base combine these embeddings using mean pooling to obtain summary-level embeddings E. The resulting embedding is a 768-dimensional vector representing the gene/GO summary.

3.3. Training Procedure

In *Genen*, we experimented with two types of GNN, namely Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT). GNNs can propagate node information within the graph structure, updating node embeddings through dynamic information flow. Both GCN and GAT models are layered, with each layer transforms node features into successively more abstract representations. These transformed features are then used to predict node categories. GCN updates node information through neighborhood aggregation, where each node aggregates the feature information of its neighbors. The specific update formula is:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{1}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the graph with added self-loops, $\tilde{D}$ is the degree matrix, and $W^{(l)}$ is the weight matrix for layer l. Here, σ denotes a nonlinear activation. Our implementation found that GCN performs best with three hidden layers.

The second GNN is GAT, a graph attention network that updates node features using an attention mechanism. GAT learns different weights to each neighbor node based on the relative importance between nodes, and dynamically aggregates neighbor node information by:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[h_i \parallel h_j]))}{\sum_{k\in\mathcal{N}(i)}\exp(\text{LeakyReLU}(\mathbf{a}^T[h_i \parallel h_k]))} \tag{2}$$

where α is a learnable parameter, and $\parallel$ denotes concatenation. LeakyReLU is a nonlinear activation function. This mechanism allows the model to focus more on relevant features. Our empirical experiments showed that GAT performs best with two hidden layers for our prediction tasks.

Additionally, we found that large hidden layer dimensions cause overfitting, so we introduced Dropout to prevent overfitting in larger hidden layer dimensions and determined the best hidden layer dimensions through cross-validation. For both GCN or GAT, using

convolutional or attention layers instead of linear layers in the output layer improved accuracy by about 1%. Cross-entropy loss was used to train these models for tasks like gene function prediction, while MSE loss was used for gene conseravation prediction..

3.4. Graph Structures and Variants

We utilize GAT and GCN on three graphs: PPI, GO graph, and GO+PPI graph. In each of these graphs, the nodes features are initialized with GeneLLM-Base embeddings [11]. The **PPI** graph is composed of gene nodes only, connected by an edge if two proteins of the associated genes have an interaction score greater than 150. In one model, the edge weights are randomly initialized and learned by the network, and in the other, the edge weights are fixed and set to the score in the PPI data ('fixed edge weights'). The **GO** graph contains two types of nodes: gene and GO. Gene nodes are connected to each other, but are connected to GO nodes if there is a known association. GO nodes are connected to each other based on the GO graph (see Section 3.1). The **GO + PPI** graph contains both gene and GO nodes, all gene-gene edges (PPI, without edge weights), gene-GO edges, and GO-GO edges.

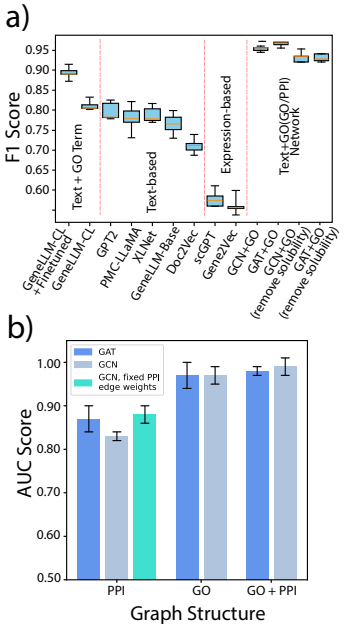

Figure 2 Performance of *Genen* in predicting the solubility of protein products of genes **a)** compared to baseline models, and **b)** using various graph structures.

4. Results

4.1. Solubility Prediction Accuracy

Similar to GeneLLM [11], we evaluated the performance of *Genen* using protein solubility prediction in aqueous media as a case study. Protein solubility in aqueous environments is crucial for functions such as transport, drug targeting, and enzyme activity [8]. Testing if a protein is soluble requires laborious and expensive protein purification and testing and sometime infeasible for many proteins [26]. GeneLLM using contrastive learning outperformed baseline methods, including task-specific models [11]; however, by using GNNs to enhance the embeddings instead of contrastive learning, *Genen* significantly surpassed GeneLLM in solubility prediction accuracy as shown in Figure 2a. This high accuracy raised concerns about potential class leakage due to the presence of terms like "soluble," "solubility," and "insolubility" in GO summaries, which might result in data leakage. To address this, we removed all GO terms containing these words from our graph. The performance of the model remained robust even after this adjustment (Figure 2a), with only a slight decrease.

4.2. Graph Structures

We evaluate the performance of Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) across various graph structures for predicting gene solubility as shown in Figure 2b. The x-axis shows three graph configurations with various constructions.

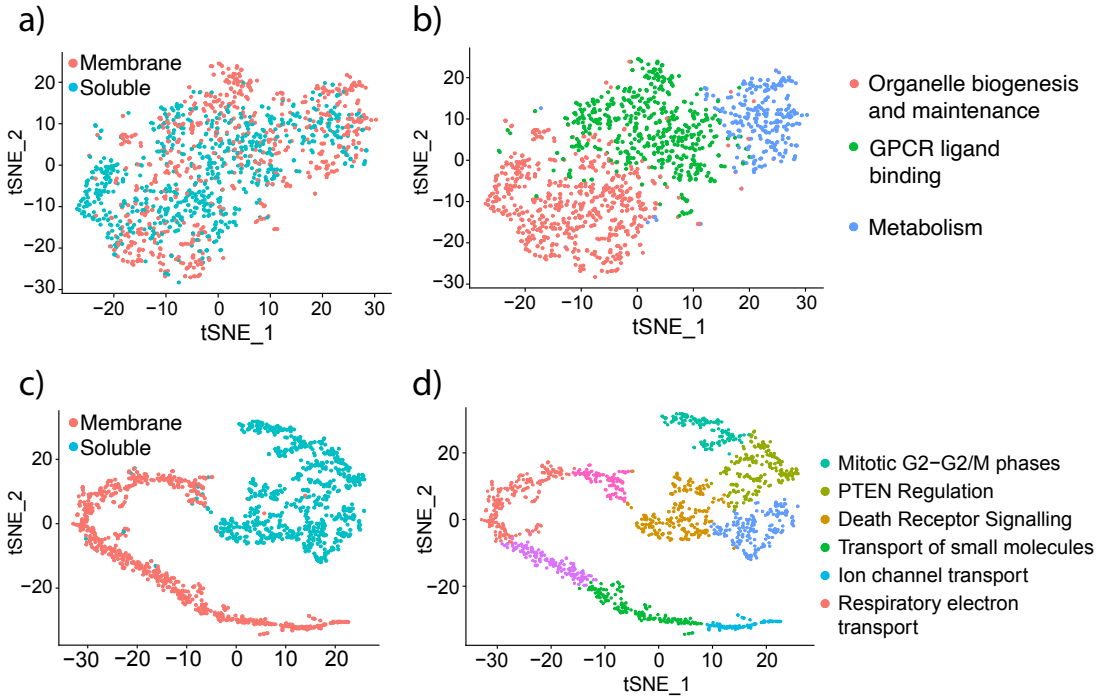

Figure 3 tSNE Clustering analysis. **a)** The initial GeneLLM-Base embeddings are shown, where each point represents a gene embedding and is colored according to its class. **b)** The same points are shown as in **a** but are labeled according to pathway enrichment analysis. **c)** and **d)** show the same analysis as **a** and **b** respectively with Genen embeddings.

For the PPI graph (Figure 2b left), the GCN model with edge weights outperforms the model without edge weights in terms of both AUC accuracy and F1 score. The GAT model also shows competitive performance, demonstrating its effectiveness in focusing on relevant features through the attention mechanism (GAT inherently has no edge weights).

For the GO graph (Figure 2b middle) containing only gene/GO edges, both GCN and GAT show excellent performance, with GAT slightly outperforming GCN. This suggests that the attention mechanism in GAT is particularly beneficial in networks where direct protein-protein interactions are absent but other forms of relational data are present.

In addition to gene/GO edges, the presence of protein-protein edges improves the performance of the GCN model (Figure 2b right). However, GAT, while achieving a slightly lower AUC, exhibited the highest F1 score among all tested configurations, indicating its superior capability in balancing precision and recall.

4.3. Interpretability

Clustering of *Genen*'s node embeddings are highly separable compared to the input embeddings from GeneLLM-Base (Figure 2c), resulting in high accuracy. Gene enrichment analysis provides further interpretability regarding *Genen*'s knowledge (Figure 2d), as seen in the emergence of more distinct clustered pathways. These provide insight to biologists who can then reason as to why a certain cluster would be likely to contain a certain class.

For example, proteins involved in death receptor signaling, such as those from the TNF receptor superfamily, are typically soluble in aqueous media because they often contain hydrophilic regions and undergo post-translational modifications [16]. For instance, Death Receptor 6 has glycosylation sites that facilitate its interaction with water molecules, making it soluble in cellular aqueous environments [16]. These structural features allow these proteins to function in signaling pathways related to apoptosis in the aqueous environments of the cytoplasm and extracellular space.

4.4. Bias

Information Propagation Infers Gene Information with 75% Data Hidden: A Potent Tool for Mitigating Knowledge Bias To investigate how well information propagation can compensate for uninformative embeddings, we randomly initialize a subset of nodes' embeddings to zero vectors in place of those generated by GeneLLM-base. These masked nodes represent the extreme case of genes with low-information embeddings (e.g., very short gene summaries). A striking finding is that even without text information for up to 75% of nodes, GNNs maintained performance levels comparable to scenarios with complete gene information (Figure 4b). We evaluated model performance for a localization task and observed a similar trend. This study underscores GNNs as a potent tool for mitigating knowledge bias.

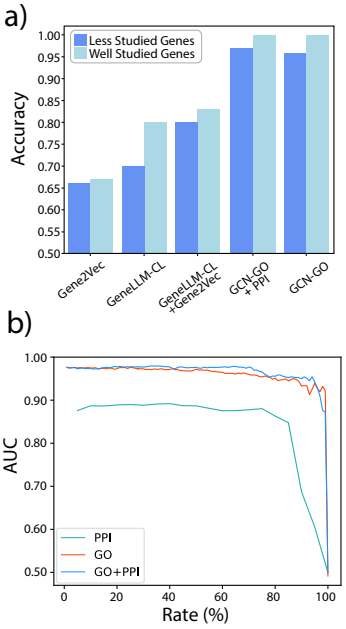

Figure 4 *Genen* performance in mitigation knowledge bias: **a)** Comparison of accuracy for less studied and well-studied genes across different models. **b)** Accuracy versus node masking rate (%) for three graph configurations, demonstrate the robustness of Genen even with significant node masking.

Information Propagation Is More Effective in Mitigating Knowledge Bias than Multimodal Approach Human biases confound our knowledge of genes. Some genes are extensively studied due to their disease associations, while many remain under-researched [20]. Studies on less-studied genes are harder to publish or fund, exacerbating knowledge bias [20]. AI models trained on such biased data further perpetuate these biases [24]. We use the gene summary length as a proxy for the information known about a gene, and separately evaluate genes with limited (less-studied) and extensive (well-studied) online information. We have previously demonstrated in GeneLLM that knowledge bias in representations derived from text could be partially mitigated by combining them with representations from non-text modalities [11]. Here we show that GNN's message passing mitigated knowledge bias more effectively than the multimodal approach for solubility prediction as shown in Figure 4b. Counterintuitively, higher accuracy was observed for less-studied genes compared to well-studied ones in the solubility task (Figure 4b). This suggests that for solubility, reliance on information propagation from neighboring annotated genes is more informative than local features of the genes themselves.

4.5. Generality

Evaluating GNN's Performance on Gene Tasks In addition to solubility prediction, our evaluation included seven gene-related tasks: Dosage Sensitivity, Chromatin State Predictions, Transcription Factor (TF) Range Prediction, TF Target Type Identification, Protein Localization, and Gene Conservation (Table 1).

Genen outperformed baseline methods in Chromatin state predictions, Protein localization, Conservation, and Dosage sensitivity (genes sensitive to changes in copy number variation). However, *Genen* performed poorly on TF-related tasks, likely due to the small number of annotated genes available for training, indicating that GNNs require larger training datasets for performance.

Model	Dosage Sensitivity	BivalentVs Lys4 Methylated	BivalentVs Non Methylated	Tf range	Tf target type	Solubility	Subcellular localization	Conservation (Pearson Corr.)
GPT2	0.83 ± 0.06	0.91 ± 0.06	0.83 ± 0.09	**0.67 ± 0.13**	0.52 ± 0.04	0.88 ± 0.01	0.91 ± 0.00	0.31 ± 0.02
Doc2Vec	0.78 ± 0.09	0.90 ± 0.08	0.79 ± 0.11	0.47 ± 0.09	0.54 ± 0.04	0.85 ± 0.02	0.85 ± 0.01	0.34 ± 0.01
PMC-LLaMA	**0.89 ± 0.04**	0.87 ± 0.03	**0.90 ± 0.09**	0.52 ± 0.45	0.06 ± 0.01	0.78 ± 0.03	0.83 ± 0.01	**0.55 ± 0.01**
XLNet	0.81 ± 0.08	0.90 ± 0.06	0.81 ± 0.06	0.61 ± 0.08	0.52 ± 0.01	0.86 ± 0.02	0.89 ± 0.01	0.40 ± 0.01
Gene2Vec	0.88 ± 0.04	0.88 ± 0.07	0.75 ± 0.06	0.56 ± 0.12	**0.58 ± 0.01**	0.60 ± 0.02	0.73 ± 0.01	0.50 ± 0.02
BERT-Base	0.85 ± 0.06	0.87 ± 0.05	0.85 ± 0.07	0.49 ± 0.11	0.53 ± 0.02	0.84 ± 0.02	0.90 ± 0.01	0.43 ± 0.01
GeneLLM	**0.89 ± 0.06**	0.87 ± 0.08	0.79 ± 0.10	0.47 ± 0.04	— ± —	0.89 ± 0.01	0.94 ± 0.03	0.53 ± 0.01
GCN	**0.89 ± 0.03**	**0.92 ± 0.01**	**0.90 ± 0.03**	0.59 ± 0.08	0.52 ± 0.03	**0.99 ± 0.02**	**0.95 ± 0.00**	**0.55 ± 0.01**

Table 1 Evaluation of *Genen*: Performance in 5-Fold AUC for gene embedding approaches across eight gene prediction tasks.

5. Conclusion

In this study, we introduce *Genen*, a model that integrates large language models (LLMs) with graph neural networks (GNNs) to enhance gene representation learning by combining textual data with relational information propagation. Through its inherent message passing mechanism, GNN propagates richer information from well-studied genes to under-studied genes. Our evaluations demonstrate that *Genen* consistently performs well across eight benchmark tasks. The model notably exhibits robustness in handling incomplete datasets, maintaining high accuracy even with significant data removal. This research emphasizes the effectiveness of information propagation in compensating for uninformative embeddings; thus, inspiring further study in the application of these methods to overcoming biased datasets that underrepresent certain groups and to additional genomic tasks. Future work will focus on expanding the applications of *Genen*, integrating additional data modalities, and further enhancing interpretability to promote more equitable AI applications in genomic research. By continuously optimizing the *Genen* model, we aim to facilitate more effective and fair scientific discoveries in various biological problem-solving contexts.

Acknowledgments

This work is supported by the National Institutes of Health under grant number R00CA248953.

References

[1] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

[2] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, and Bo Wang. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv*, 2023.

[3] Christian Dallago, Konstantin Schütze, Michael Heinzinger, Tobias Olenyi, Maria Littmann, Amy X Lu, Kevin K Yang, Seonwoo Min, Sungroh Yoon, James T Morton, et al. Learned embeddings from deep learning to visualize and predict protein sets. *Current Protocols*, 1(5):e113, 2021.

[4] Joseph M. de Guia, Madhavi Devaraj, and Carson K. Leung. Deepgx: deep learning using gene expression for cancer classification. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '19, page 913–920, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368681. doi: 10.1145/3341161.3343516. URL https://doi.org/10.1145/3341161.3343516.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[6] Jingcheng Du, Peilin Jia, YuLin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics*, 20(1):82, Feb 2019. ISSN 1471-2164. doi: 10.1186/s12864-018-5370-x.

[7] Jingcheng Du, Peilin Jia, Yulin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics*, 20(1):82–, 2019. ISSN 14712164. doi: 10.1186/s12864-018-5370-x. URL https://doi.org/10.1186/s12864-018-5370-x.

[8] Michael R Dyson, Rajika L Perera, S Paul Shadbolt, Lynn Biderman, Krystyna Bromek, Natalia V Murzina, and John McCafferty. Identification of soluble protein fragments by gene fragmentation and genetic selection. *Nucleic acids research*, 36(9): e51–e51, 2008.

[9] S Gatzemeier. Ai bias: Where does it come from and what can we do about it. *Data Science W231-Behind the Data: Humans and Values*, 2021.

[10] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model

pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779, 2020. URL https://arxiv.org/abs/2007.15779.

[11] Ala Jararweh, Oladimeji Macaulay, David Arredondo, Olufunmilola Oyebamiji, Luis E Tafoya, Kushal Virupakshappa, and Avinash Sahu. Unveiling zero shot prediction for gene attributes through interpretable AI. In *ICLR 2024 Workshop on Machine Learning for Genomics Explorations*, 2024. URL https://openreview.net/forum?id=DtdLDKe32W.

[12] Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models, 2023.

[13] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.

[14] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[16] You-Take Oh and Shi-Yong Sun. Regulation of cancer metastasis by trail/death receptor signaling. *Biomolecules*, 11(44):499, April 2021. ISSN 2218-273X. doi: 10.3390/biom11040499.

[17] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *preprint*, 2018. Technical report, OpenAI.

[18] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. URL https://dcmpx.remotevs.com/net/cloudfront/d4mucfpksywv/SL/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[19] Laleh Seyyed-Kalantari, Haoran Zhang, Matthew BA McDermott, Irene Y Chen, and Marzyeh Ghassemi. Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations. *Nature medicine*, 27(12): 2176–2182, 2021.

[20] Thomas Stoeger, Martin Gerlach, Richard I. Morimoto, and Luís A. Nunes Amaral. Large-scale investigation of the reasons why potentially important genes are ignored.

PLOS Biology, 16(9):e2006643, September 2018. ISSN 1545-7885. doi: 10.1371/journal.pbio.2006643.

[21] Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, pages 1–9, 2023.

[22] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *CoRR*, abs/1910.10699, 2019. URL http://arxiv.org/abs/1910.10699.

[23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[24] M. Viswanathan, C.D. Patnode, N.D. Berkman, et al. Assessing the risk of bias in systematic reviews of health care interventions. https://www.ncbi.nlm.nih.gov/books/NBK519366/, Dec 13 2017.

[25] Kerstin N Vokinger, Stefan Feuerriegel, and Aaron S Kesselheim. Mitigating bias in machine learning for medicine. *Communications medicine*, 1(1):25, 2021.

[26] Chao Wang and Quan Zou. Prediction of protein solubility based on sequence physicochemical patterns and distributed representation information with deepsolue. *BMC biology*, 21(1):1–11, 2023.

[27] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-llama: Towards building open-source language models for medicine, 2023.

[28] Chunlei Wu, Ian MacLeod, and Andrew I Su. Biogps and mygene. info: organizing online, gene-centric information. *Nucleic acids research*, 41(D1):D561–D565, 2013.

[29] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

[30] Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Jiazhang Lian, Qiang Zhang, and Huajun Chen. Ontoprotein: Protein pretraining with gene ontology embedding, 2022.

Appendix A. GNN enhanced F1 Scores

Model	Dosage Sensitivity	BivalentVs Lys4 Methylated	BivalentVs Non Methylated	Tf range	Tf target type	Solubility	Subcellular localization	Conservation (Pearson Corr.)
GPT2	0.73 ± 0.04	0.86 ± 0.04	0.80 ± 0.11	0.67 ± 0.04	0.17 ± 0.01	0.80 ± 0.02	0.77 ± 0.01	0.31 ± 0.02
Doc2Vec	0.73 ± 0.06	0.84 ± 0.06	0.84 ± 0.05	0.57 ± 0.04	0.20 ± 0.02	0.71 ± 0.03	0.69 ± 0.02	0.34 ± 0.01
PMC-LLaMA	0.86 ± 0.05	0.77 ± 0.04	0.83 ± 0.08	0.60 ± 0.04	0.07 ± 0.01	0.78 ± 0.03	0.69 ± 0.01	**0.55 ± 0.01**
XLNet	0.74 ± 0.06	0.84 ± 0.06	0.82 ± 0.08	0.64 ± 0.06	0.12 ± 0.01	0.79 ± 0.02	0.76 ± 0.01	0.40 ± 0.01
Gene2Vec	0.84 ± 0.04	0.84 ± 0.06	0.75 ± 0.06	**0.70 ± 0.06**	0.19 ± 0.01	0.56 ± 0.02	0.54 ± 0.02	0.50 ± 0.02
BERT-Base	0.76 ± 0.09	0.83 ± 0.06	0.76 ± 0.09	0.64 ± 0.06	0.16 ± 0.01	0.77 ± 0.02	0.76 ± 0.01	0.43 ± 0.01
GeneLLM	**0.87 ± 0.06**	0.86 ± 0.09	0.82 ± 0.08	0.74 ± 0.07	**0.49 ± 0.04**	0.89 ± 0.01	0.83 ± 0.01	0.53 ± 0.01
GCN	0.82 ± 0.04	**0.91 ± 0.03**	**0.86 ± 0.03**	0.53 ± 0.04	0.40 ± 0.01	**0.95 ± 0.01**	**0.86 ± 0.01**	0.55 ± 0.01

Table A.1 Evaluation of *Genen*: Performance in 5-Fold F1 for gene embedding approaches across eight gene prediction tasks.

Appendix B. Subcellular Localization

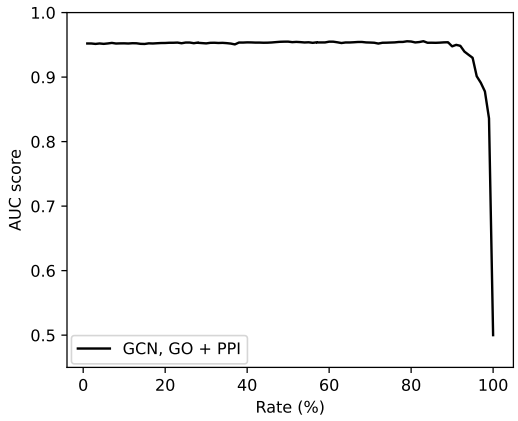

Figure B.1 AUC vs masking rate (as in main text Figure 4b) for the task of predicting subcellular localization, i.e., whether a gene product localizes to the cytoplasm, nucleus, or cell membrane.

Appendix C. Baselines

We perform an extensive evaluation of our model against various baseline models. These baselines encompass a broad range of representation learning techniques, trained either on gene co-expression transcriptome data or textual data. Below, we provide a detailed description of the baselines used:

- Majority Classifier: The classifier predicts the most common class for all genes, illustrating the dataset's distribution and the difficulty of the problem.

- scGPT [2]: A transformer-based foundation model model with multi-head attention designed for gene and cell embeddings, pre-trained on 33 million normal human cells. We extract gene embeddings from their largest pre-trained model, whole-human scGPT.
- Gene2Vec [6]: Neural network based on Word2Vec[15] trained a wide range of gene co-expression datasets, capturing gene functions and interactions.
- Doc2Vec [14]: A neural network model building on Word2Vec [15] creates embeddings for bodies of text. We use an embedding size of 50, maximum distance between current and predicted word within a sentence of 2, all words with a total frequency of 1, and 40 training epochs.
- XLNet [29]: An autoregressive transformer pretraining method. We obtain gene embeddings by utilizing the 12-layer xlnet-base-cased model with CLS pooling.
- PMC-LLaMA [27]: A LLaMA-based foundation model [23] pre-trained on biomedical text and optimized for medical applications. We obtain embeddings from PMC-LLaMA using prompt-based last token pooling with the prompt *"This sentence: text means in one word:[CLS]"*, utilizing the contextualized embedding of the last token, which is the CLS token added by the tokenizer after the colon [12].
- GPT-2 [18]: An open-source transformer-based foundation language model trained on the WebText dataset [18]. Embeddings are generated from the encoder of GPT2 by performing CLS pooling.

The baselines generate general summary embeddings. Consequently, we tailor our analysis for specific downstream tasks by augmenting these embeddings with a Linear/Logistic Regression (LR) model.

Appendix D. Evaluation Metrics

To evaluate the performance of our models in node classification tasks, we use AUC and F1 scores as our primary metrics. This choice of metrics is crucial for several reasons. Firstly, F1 score, being the harmonic mean of precision and recall, is particularly useful in dealing with imbalanced data. In bioinformatics or social network analysis, it is common to have an imbalanced class distribution, where some classes may have far more nodes than other classes.. It ensures that both the retrieval of relevant instances (recall) and the precision of these retrievals are considered, offering a balanced view of model performance across different classes. Secondly, AUC provides an aggregate measure of model performance at various threshold settings, making it a robust metric against different classification thresholds. This is vital as it reflects the model's ability to discriminate between the node categories with high reliability. Together, AUC and F1 offer comprehensive insights into the performance of our models, addressing both the balance between precision and recall and the ability to distinguish between classes under varied threshold scenarios. These metrics, therefore, facilitate a deeper understanding of the models' predictive capabilities and ensure that we develop systems that are both effective and equitable in their predictive performance.